MT7628 DATASHEET

Overview

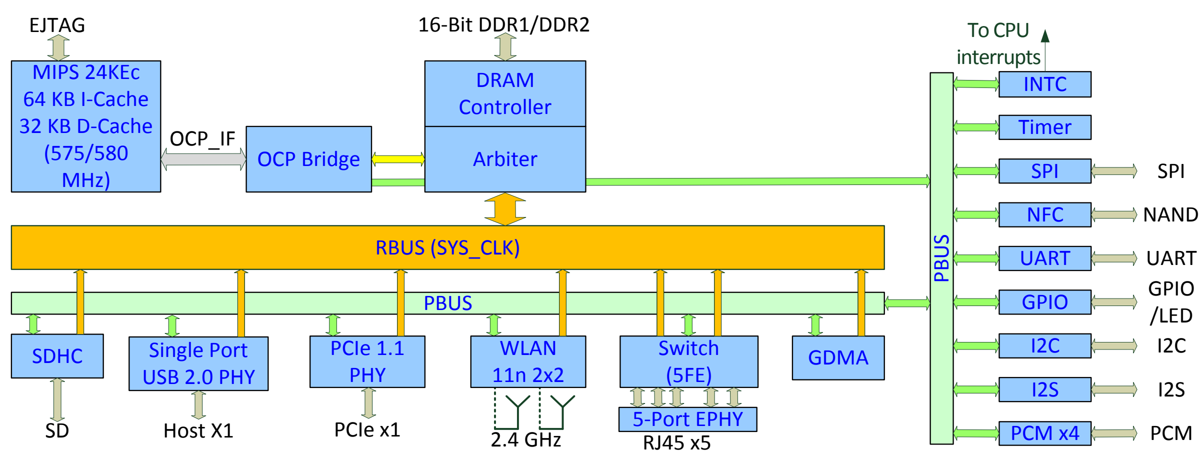
The MT7628 router-on-a-chip includes an 802.11n MAC and baseband, a 2.4 GHz radio and FEM, a 575/580 MHz MIPS® 24K™ CPU core, a 5-port 10/100 fast ethernet switch. The MT7628 includes everything needed to build an AP router from a single chip. The embedded high performance CPU can process advanced applications effortlessly, such as routing, security and VoIP. The MT7628 also includes a selection of interfaces to support a variety of applications, such as a USB port for accessing external storage.

Applications:
- Routers
- NAS devices
- Dual band concurrent routers

Features

- Embedded MIPS24KEc (575/580 MHz) with 64 KB I-Cache and 32 KB D-Cache
- 2T2R 2.4 GHz with 300 Mbps PHY data rate
- Legacy 802.11b/g and HT 802.11n modes
- 20/40 MHz channel bandwidth
- Reverse Data Grant (RDG)
- Maximal Ratio Combining (MRC)
- Space Time Block Coding (STBC)
- MCM 8 Mbytes DDR1 KGD (MT7628KN)
- 16-bit DDR1/2 up to 128/256 Mbytes (MT7628AN/KN)
- SPI/SD-XC/eMMC
- x1 USB 2.0 Host, x1 PCIe Root Complex
- 5-port 10/100 FE PHY
- Internet Of Thing
- An optimized PMU
- Green AP
  - Intelligent Clock Scaling (exclusive)
  - DDRII: ODT off, Self-refresh mode
- I2C, I2S, SPI, PCM, UART, JTAG, GPIO
- 16 Multiple BSSID
- WEP64/128, TKIP, AES, WPA, WPA2, WAPI
- QoS: WMM, WMM-PS
- WPS: PBC, PIN
- Voice Enterprise: 802.11k+r
- AP Firmware: Linux 2.6 SDK, eCOS with IPv6

Functional Block Diagram



Ordering Information

Part Number	Package (Green/RoHS Compliant)
MT7628AN	DR-QFN 156 pin (12 mm x 12 mm)
MT7628KN	DR-QFN 120 pin (10 mm x 10 mm)

Table of Contents

Table of Figures

List of Tables

1. Main Features

The following table covers the main features offered by the MT7628KN and MT7628AN. Overall, the MT7628KN supports the requirements of an entry-level AP/router, while the more advanced MT7628AN supports a number of interfaces together with a large maximum RAM capacity.

Features	MT7628KN	MT7628AN
CPU	MIPS24KEc (575/580 MHz)	MIPS24KEc (575/580 MHz)
Total DMIPs	580 x 1.6 DMIPs	580 x 1.6 DMIPs
I-Cache, D-Cache	64 KB, 32 KB	64 KB, 32 KB
L2 Cache	n/a	n/a
Memory		
DRAM Device width support	16 bits	16 bits
DDR1	64 Mb (MCM), 193 MHz	2 Gb, 193 MHz
DDR2	n/a	2 Gb, 193 MHz
SPI Flash	3B addr mode (max 128Mbit) 4B addr mode (max 512Mbit)	3B addr mode (max 128Mbit) 4B addr mode (max 512Mbit)
SD	n/a	SD-XC (class 10)
RF	2T2R 802.11n 2.4 GHz	2T2R 802.11n 2.4 GHz
PCIe	1	1
USB 2.0	1	1
Switch	5p FE SW	5p FE SW
I2S	1	1
PCM	1	1
I2C	1	1
UART	2 (Lite)	2 (Lite)
JTAG	1	1
Package	DR-QFN120- 10 mm x 10 mm	DR-QFN156- 12 mm x 12 mm

Table 1-1 Main Features

2. Pins

2.1 MT7628AN DR-QFN (12 mm x 12 mm) 156-Pin Package Diagram

2.1.1 Up-left side

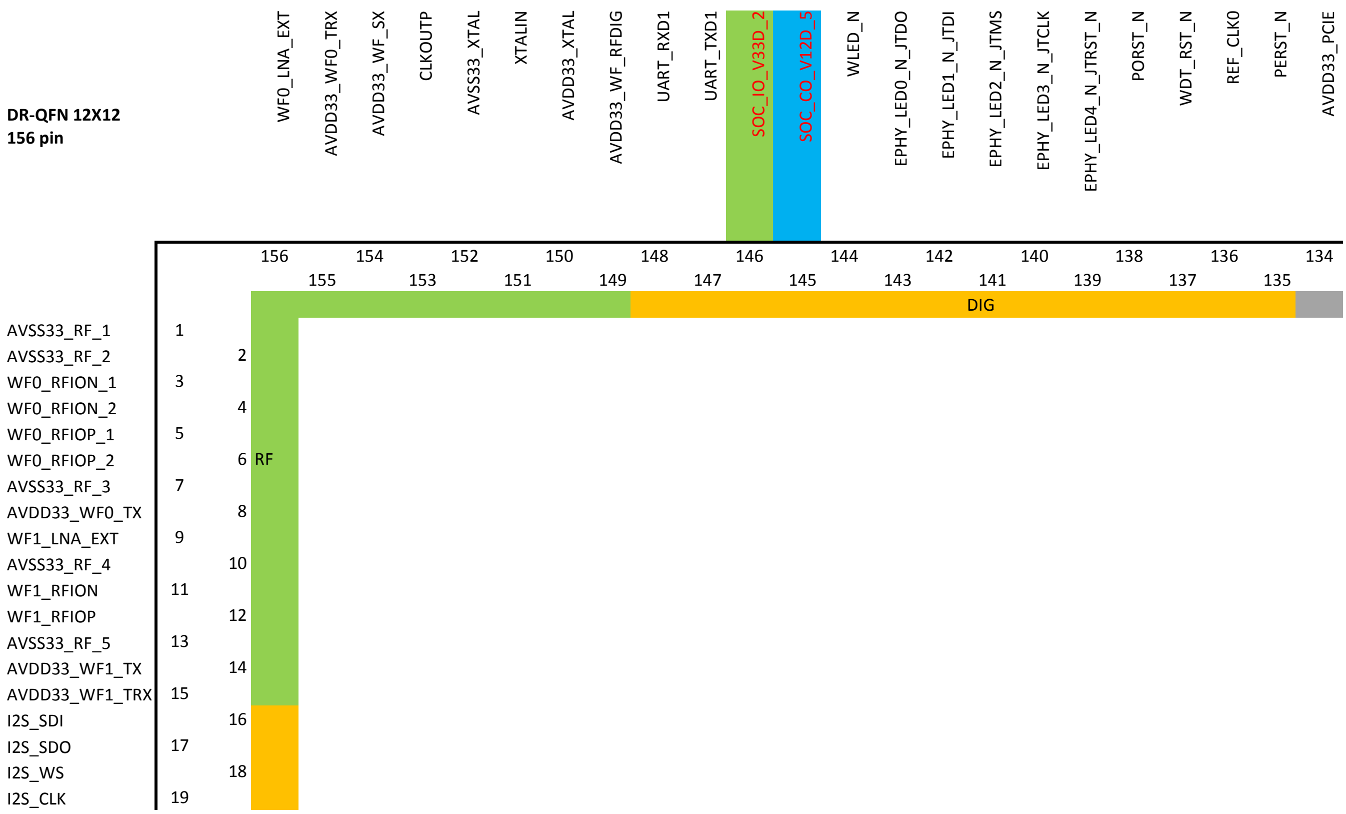


Figure 2-1 MT7628AN DR-QFN Pin Diagram (up-left view)

2.1.2 Down-left side

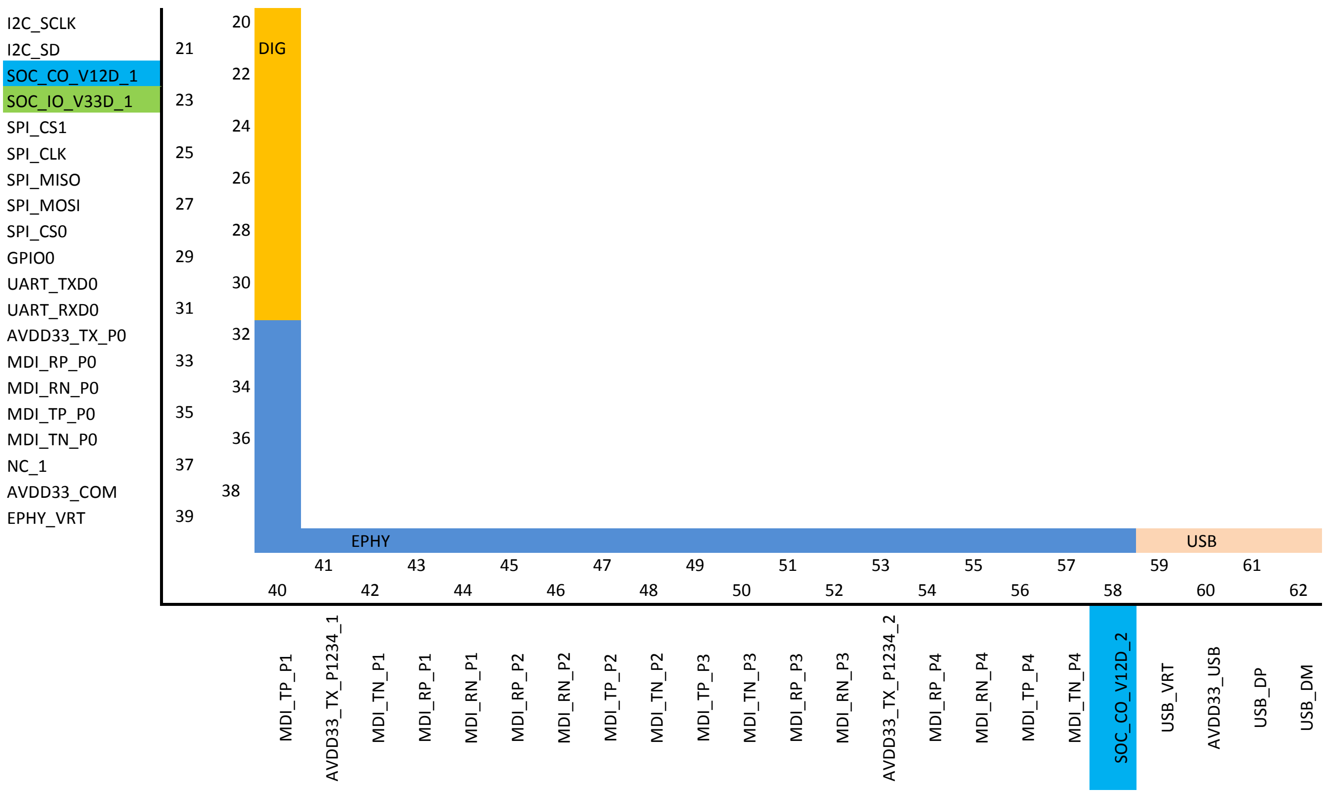


Figure 2-2 MT7628AN DR-QFN Pin Diagram (down-left view)

2.1.3 Down-right side

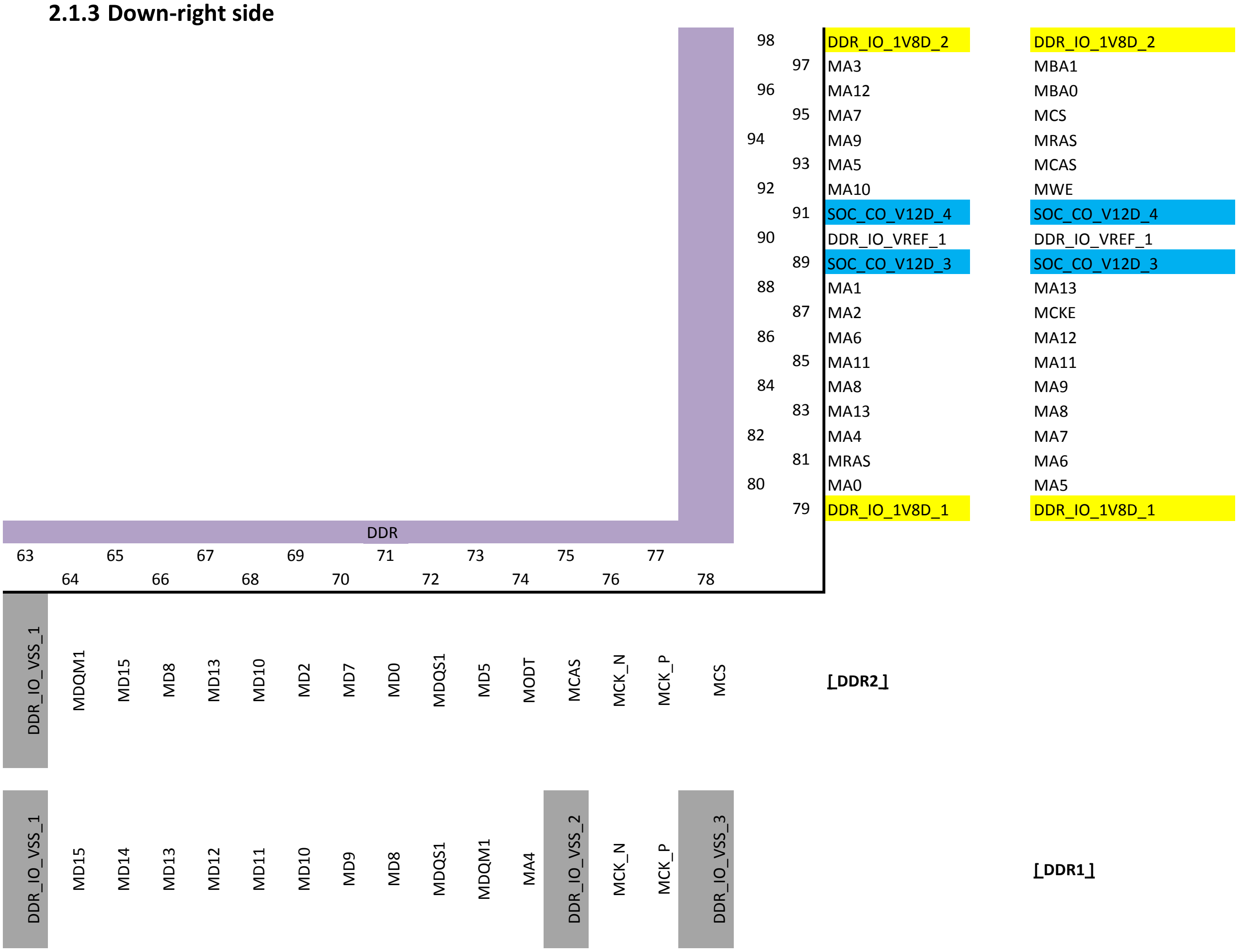


Figure 2-3 MT7628AN DR-QFN Pin Diagram (down-right view)

Note: DR-QFN support DDR1 and DDR2 pin shuffle depend on the bootstrap.

2.1.4 Up-right side

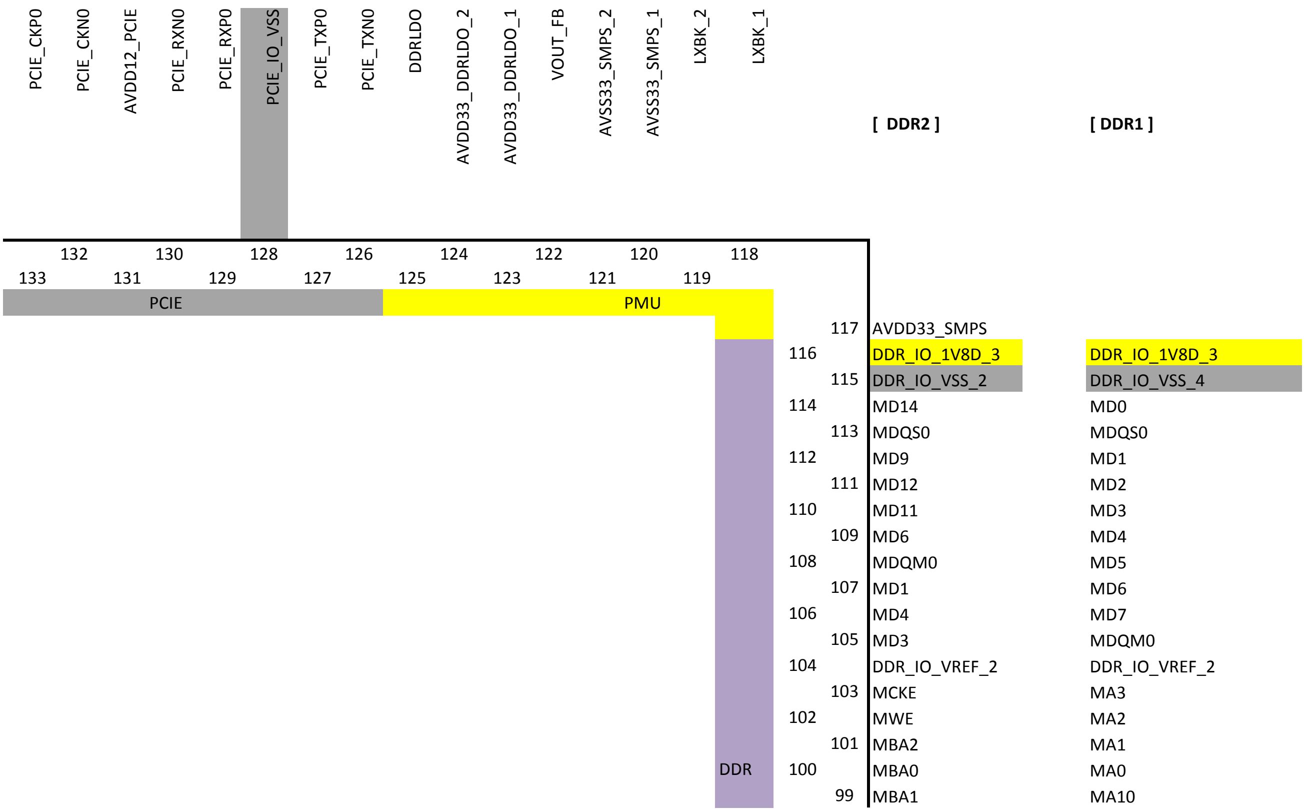


Figure 2-4 MT7628AN DR-QFN Pin Diagram (up-right view)

2.1.5 Pin Description

Pins	Name	Type	Driv.	Description
RF				
3,4	WF0_RFION_1 WF0_RFION_2	A		WF0 main path RF I/O
5,6	WF0_RFIOP_1 WF0_RFIOP_2	A		WF0 main path RF I/O
11	WF1_RFION	A		WF1 main path RF I/O
12	WF1_RFIOP	A		WF1 main path RF I/O
9	WF1_LNA_EXT	A		WF1 aux. path LNA input
156	WF0_LNA_EXT	A		WF0 aux. path LNA input
151	XTALIN	I		Crystal oscillator input
153	CLKOUTP	O		XO reference clock output
150	AVDD33_XTAL	P		3.3V XTAL Power Supply Pin
152	AVSS33_XTAL	G		3.3V XTAL Ground Pin
8	AVDD33_WF0_TX	P		3.3V RF Channel 0 Suppoly Power
14	AVDD33_WF1_TX	P		3.3V RF Channel 1 Suppoly Power
15	AVDD33_WF1_TRX	P		1.65V to 3.3V RF Channel 1 Suppoly Power
149	AVDD33_WF_RFDIG	P		1.65V to 3.3V RF DIG and AFE Suppoly Power
154	AVDD33_WF_SX	P		1.65V to 3.3V RF Supply Power
155	AVDD33_WF0_TRX	P		1.65V to 3.3V RF Channel 0 Suppoly Power
1,2 7,10,13	AVSS33_RF	G		3.3V RF Shielding Ground Pin
WLAN LED				
144	WLED_N	O	4 mA	WLAN Activity LED
UART0 Lite				
31	UART_RXD0	I	4 mA	UART0 Lite RXD
30	UART_TXD0	O, IPD	4 mA	UART0 Lite TXD
UART1 Lite				
147	UART_TXD1	O, IPU	4 mA	UART1 Lite TXD
148	UART_RXD1	I	4 mA	UART1 Lite RXD
I2S				
16	I2S_SDI	O	4 mA	I2S data input
17	I2S_SDO	I/O, IPD	4 mA	I2S data output
18	I2S_WS	O	4 mA	I2S word select
19	I2S_CLK	I/O	4 mA	I2S clock
I2C				
21	I2C_SD		4 mA	I2C Data

Pins	Name	Type	Driv.	Description
20	I2C_SCLK	I/O	4 mA	I2C Clock
SPI				
26	SPI_MISO	I/O	4 mA	SPI Master input/Slave output
27	SPI_MOSI	I/O, IPD	4 mA	SPI Master output/Slave input
25	SPI_CLK	O, IPU	4 mA	SPI clock
28	SPI_CS0	O	4 mA	SPI chip select0
24	SPI_CS1	O, IPD	4 mA	SPI chip select1
GPIO				
29	GPIO0	I/O, IPD	4 mA	General Purpose I/O
5-Port EPHY				
143	EPHY_LED0 _N_JTDO	I/O	4 mA	10/100 PHY Port #0 activity LED, JTAG_TDO
142	EPHY_LED1 _N_JTDI	I/O	4 mA	10/100 PHY Port #1 activity LED, JTAG_TDI
141	EPHY_LED2 _N_JTMS	I/O	4 mA	10/100 PHY Port #2 activity LED, JTAG_TMS
140	EPHY_LED3 _N_JTCLK	I/O	4 mA	10/100 PHY Port #3 activity LED, JTAG_CLK
139	EPHY_LED4 _N_JTRST_N	I/O,	4 mA	10/100 PHY Port #4 activity LED, JTAG_TRST_N
39	EPHY_VRT	A		Connect to an external resistor to provide accurate bias current
33	MDI_RP_P0	A		10/100 PHY Port #0 RXN
34	MDI_RN_P0	A		10/100 PHY Port #0 RXP
35	MDI_TP_P0	A		10/100 PHY Port #0 TXN
36	MDI_TN_P0	A		10/100 PHY Port #0 TXP
40	MDI_TP_P1	A		10/100 PHY Port #1 RXN
42	MDI_TN_P1	A		10/100 PHY Port #1 RXP
43	MDI_RP_P1	A		10/100 PHY Port #1 TXN
44	MDI_RN_P1	A		10/100 PHY Port #1 TXP
45	MDI_RP_P2	A		10/100 PHY Port #2 RXN
46	MDI_RN_P2	A		10/100 PHY Port #2 RXP
47	MDI_TP_P2	A		10/100 PHY Port #2 TXN
48	MDI_TN_P2	A		10/100 PHY Port #2 TXP
49	MDI_TP_P3	A		10/100 PHY Port #3 RXN
50	MDI_TN_P3	A		10/100 PHY Port #3 RXP
51	MDI_RP_P3	A		10/100 PHY Port #3 TXN
52	MDI_RN_P3	A		10/100 PHY Port #3 TXP
54	MDI_RP_P4	A		10/100 PHY Port #4 RXN
55	MDI_RN_P4	A		10/100 PHY Port #4 RXP
56	MDI_TP_P4	A		10/100 PHY Port #4 TXN

Pins	Name	Type	Driv.	Description
57	MDI_TN_P4	A		10/100 PHY Port #4 TXP
32	AVDD33_TX_P0	P		3.3V Supply Power for P0
38	AVDD33_COM	P		3.3V Supply Power for EPHY COM
41, 53	AVDD33_TX_P1234_1 AVDD33_TX_P1234_2	P		3.3V Supply Power for P1 ~ P4
Misc.				
136	REF_CLKO	O, IPD	4 mA	Reference Clock Ouptut
138	PORST_N	I, IPU	4 mA	Power on reset
137	WDT_RST_N	O	4 mA	Watchdog timeout reset
USB PHY				
60	AVDD33_USB	P		3.3 V USB PHY analog power supply
59	USB _VRT	I/O		Connect to an external 5.1 kΩ resistor for band-gap reference circuit
62	USB_DM	I/O		USB Port0 data pin Data-
61	USB _DP	I/O		USB Port0 data pin Data+
PCIe PHY				
135	PERST_N	O, IPD	4mA	PCIe device reset
131	AVDD12_PCIE	P		1.2 V PCIE PHY digital power supply
134	AVDD33_PCIE	P		3.3 V USB PHY analog power supply
128	PCIE_IO_VSS	P		PCIE PHY Ground Pin
133	PCIE_CKP0	I/O		External reference clock output (positive)
132	PCIE_CKN0	I/O		External reference clock output (negative)
127	PCIE_TXP0	I/O		PCIe0 differential transmit TX -
126	PCIE_TXN0	I/O		PCIe0 differential transmit TX -
129	PCIE_RXP0	I/O		PCIe0 differential receiver RX -
130	PCIE_RXN0	I/O		PCIe0 differential receiver RX -
DDR2				
65	MD15	I/O	8 mA	DDR2 Data bit #15
114	MD14	I/O	8 mA	DDR2 Data bit #14
67	MD13	I/O	8 mA	DDR2 Data bit #13
111	MD12	I/O	8 mA	DDR2 Data bit #12
110	MD11	I/O	8 mA	DDR2 Data bit #11
68	MD10	I/O	8 mA	DDR2 Data bit #10
112	MD9	I/O	8 mA	DDR2 Data bit #9
66	MD8	I/O	8 mA	DDR2 Data bit #8
70	MD7	I/O	8 mA	DDR2 Data bit #7
109	MD6	I/O	8 mA	DDR2 Data bit #6

Pins	Name	Type	Driv.	Description
73	MD5	I/O	8 mA	DDR2 Data bit #5
106	MD4	I/O	8 mA	DDR2 Data bit #4
105	MD3	I/O	8 mA	DDR2 Data bit #3
69	MD2	I/O	8 mA	DDR2 Data bit #2
107	MD1	I/O	8 mA	DDR2 Data bit #1
71	MD0	I/O	8 mA	DDR2 Data bit #0
83	MA13	O	8 mA	DDR2 Address bit #13
96	MA12	O	8 mA	DDR2 Address bit #12
85	MA11	O	8 mA	DDR2 Address bit #11
92	MA10	O	8 mA	DDR2 Address bit #10
94	MA9	O	8 mA	DDR2 Address bit #9
84	MA8	O	8 mA	DDR2 Address bit #8
95	MA7	O	8 mA	DDR2 Address bit #7
86	MA6	O	8 mA	DDR2 Address bit #6
93	MA5	O	8 mA	DDR2 Address bit #5
82	MA4	O	8 mA	DDR2 Address bit #4
97	MA3	O	8 mA	DDR2 Address bit #3
87	MA2	O	8 mA	DDR2 Address bit #2
88	MA1	O	8 mA	DDR2 Address bit #1
80	MA0	O	8 mA	DDR2 Address bit #0
101	MBA2	O	8 mA	DDR2 MBA #2
99	MBA1	O	8 mA	DDR2 MBA #1
100	MBA0	O	8 mA	DDR2 MBA #0
74	MODT	O	8 mA	DDR2 ODT
81	MRAS	O	8 mA	DDR2 MRAS_N
75	MCAS	O	8 mA	DDR2 MCAS_N
102	MWE	O	8 mA	DDR2 MWE_N
77	MCK_P	O	8 mA	DDR2 MCK_P
76	MCK_N	O	8 mA	DDR2 MCK_N
64	MDQM1	O	8 mA	DDR2 MDQM#1
108	MDQM0	O	8 mA	DDR2 MDQM#0
78	MCS	O	8 mA	DDR2 MCS
72	MDQS1	I/O	8 mA	DDR2 MDQS#1
113	MDQS0	I/O	8 mA	DDR2 MDQS#0
103	MCKE	O	8 mA	DDR2 MCKE
63 115	DDR_IO_VSS_1 DDR_IO_VSS_2	G		DDR IO Ground pins

Pins	Name	Type	Driv.	Description
79 98 116	DDR_IO_1V8D_1 DDR_IO_1V8D_2 DDR_IO_1V8D_3	P		DDR io Supply power
90 104	DDR_IO_VREF_1 DDR_IO_VREF_2	A		DDR reference voltage
DDR1				
64	MD15	I/O	8 mA	DDR1 Data bit #15
65	MD14	I/O	8 mA	DDR1 Data bit #14
66	MD13	I/O	8 mA	DDR1 Data bit #13
67	MD12	I/O	8 mA	DDR1 Data bit #12
68	MD11	I/O	8 mA	DDR1 Data bit #11
69	MD10	I/O	8 mA	DDR1 Data bit #10
70	MD9	I/O	8 mA	DDR1 Data bit #9
71	MD8	I/O	8 mA	DDR1 Data bit #8
106	MD7	I/O	8 mA	DDR1 Data bit #7
107	MD6	I/O	8 mA	DDR1 Data bit #6
108	MD5	I/O	8 mA	DDR1 Data bit #5
109	MD4	I/O	8 mA	DDR1 Data bit #4
110	MD3	I/O	8 mA	DDR1 Data bit #3
111	MD2	I/O	8 mA	DDR1 Data bit #2
112	MD1	I/O	8 mA	DDR1 Data bit #1
114	MD0	I/O	8 mA	DDR1 Data bit #0
88	MA13	O	8 mA	DDR1 Address bit #13
86	MA12	O	8 mA	DDR1 Address bit #12
85	MA11	O	8 mA	DDR1 Address bit #11
99	MA10	O	8 mA	DDR1 Address bit #10
84	MA9	O	8 mA	DDR1 Address bit #9
83	MA8	O	8 mA	DDR1 Address bit #8
82	MA7	O	8 mA	DDR1 Address bit #7
81	MA6	O	8 mA	DDR1 Address bit #6
80	MA5	O	8 mA	DDR1 Address bit #5
74	MA4	O	8 mA	DDR1 Address bit #4
103	MA3	O	8 mA	DDR1 Address bit #3
102	MA2	O	8 mA	DDR1 Address bit #2
101	MA1	O	8 mA	DDR1 Address bit #1
100	MA0	O	8 mA	DDR1 Address bit #0
97	MBA1	O	8 mA	DDR1 MBA #1

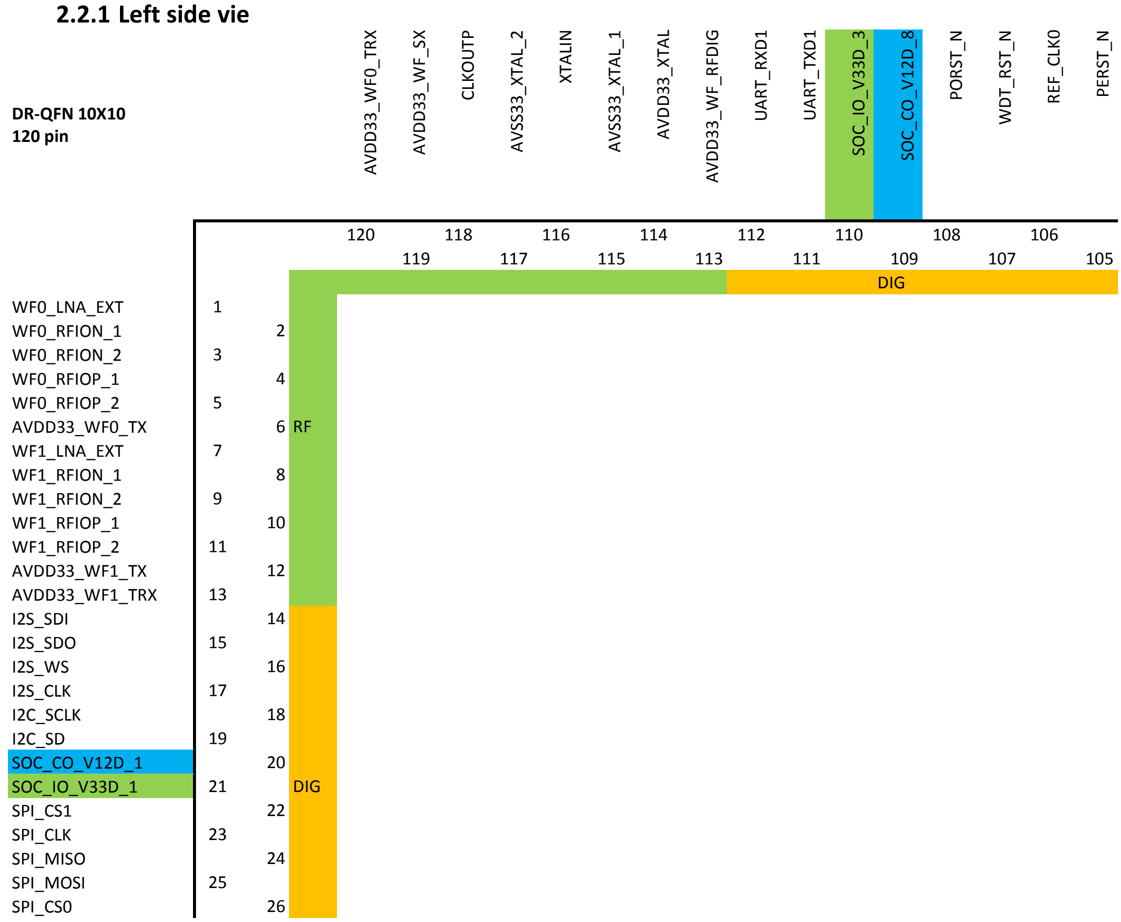
Pins	Name	Type	Driv.	Description
96	MBA0	O	8 mA	DDR1 MBA #0
94	MRAS	O	8 mA	DDR1 MRAS_N
93	MCAS	O	8 mA	DDR1 MCAS_N
92	MWE	O	8 mA	DDR1 MWE_N
77	MCK_P	O	8 mA	DDR1 MCK_P
76	MCK_N	O	8 mA	DDR1 MCK_N
73	MDQM1	O	8 mA	DDR1 MDQM#1
105	MDQM0	O	8 mA	DDR1 MDQM#0
95	MCS	O	8 mA	DDR1 MCS
72	MDQS1	I/O	8 mA	DDR1 MDQS#1
113	MDQS0	I/O	8 mA	DDR1 MDQS#0
87	MCKE	O	8 mA	DDR1 MCKE
63 75 78 115	DDR_IO_VSS_1 DDR_IO_VSS_2 DDR_IO_VSS_3 DDR_IO_VSS_4	G		DDR IO Ground pins
79 98 116	DDR_IO_1V8D_1 DDR_IO_1V8D_2 DDR_IO_1V8D_3	P		DDR IO Supply power
90 104	DDR_IO_VREF_1 DDR_IO_VREF_2	A		DDR reference voltage
PMU				
118 119	LXBK_1 LXBK_2	O		Buck Switching node
122	VOUT_FB	A		Buck vout feedback pin
117	AVDD33_SMPS	P		Buck 3.3V Supply power
120 121	AVSS33_SMPS_1 AVSS33_SMPS_2	G		Buck Gound pin
123 124	AVDD33_DDRLDO_1 AVDD33_DDRLDO_2	G		DDRLDO 3.3V Supply power
125	DDRLDO	O		DDRLDO 1.8V/2.5V output voltage
Power				
23 146	SOC_IO_V33D_1 SOC_IO_V33D_2	P		3.3 V digital I/O power supply
22 58 89 91 145	SOC_CO _V12D_1 SOC_CO _V12D_2 SOC_CO _V12D_3 SOC_CO _V12D_4 SOC_CO _V12D_5	P		1.2 V digital core power supply
EPAD	GND	G		Ground pin

Pins	Name	Type	Driv.	Description
NC				
37	NC_1	NC		No connected
Total: 156 pins				

Note:
- IPD : Internal pull-down
- IPU : Internal pull-up
- I : Input
- O : Output
- IO : Bi-directional
- P : Power
- G : Ground
- NC : Not connected

2.2 MT7628KN DR-QFN (10 mm x 10 mm) 120-Pin Package Diagram

2.2.1 Left side vie



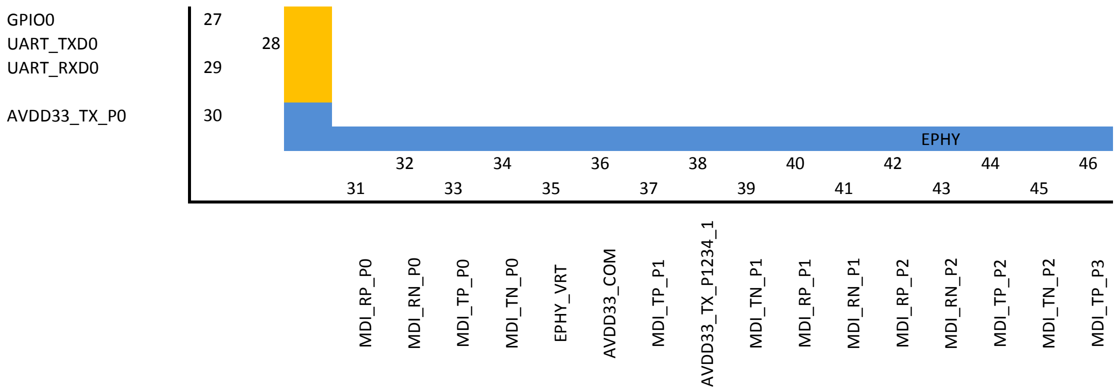


Figure 2-5 MT7628KN DR-QFN Pin Diagram (left view)

2.2.2 Right side view

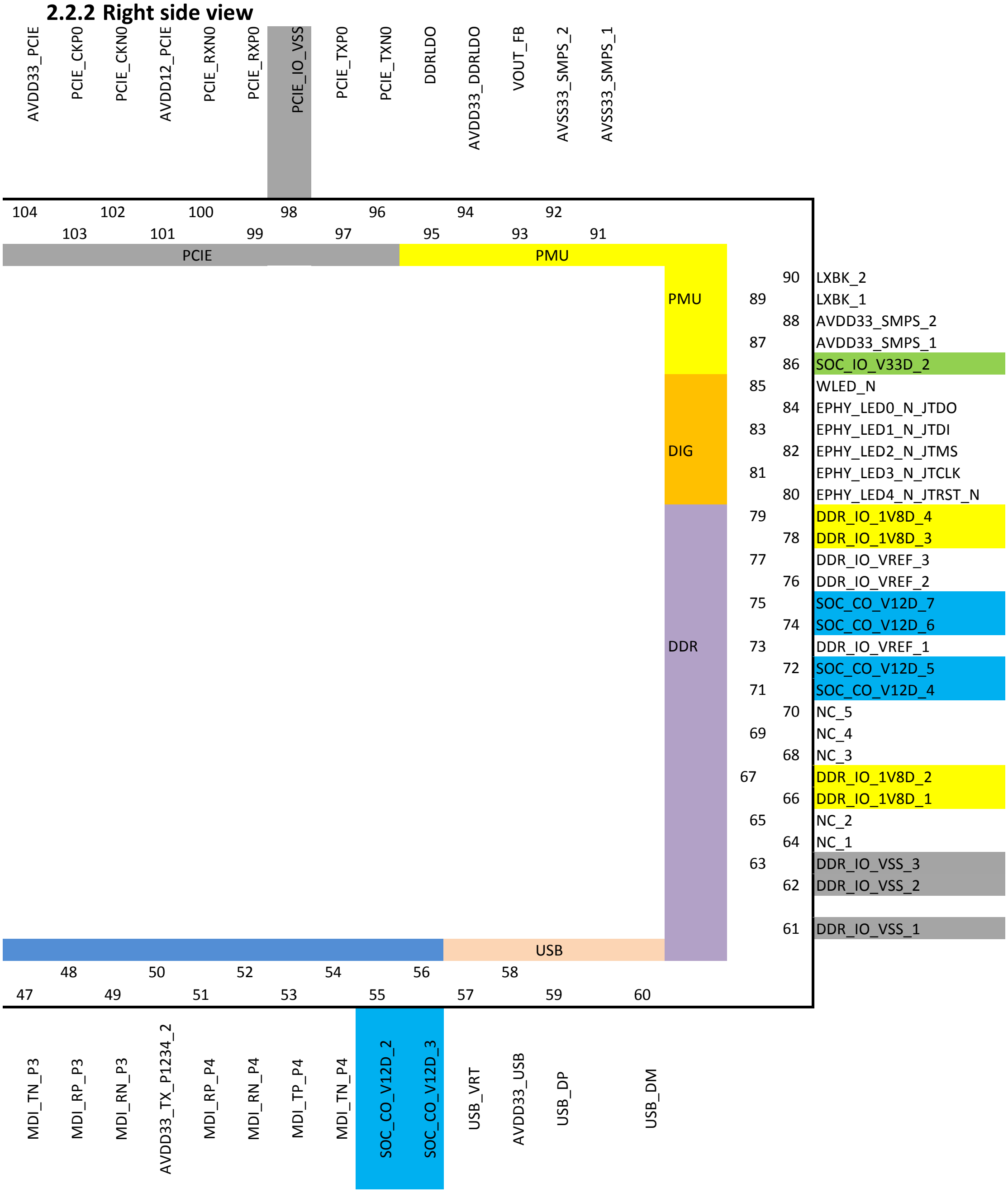


Figure 2-6 MT7628KN DR-QFN Pin Diagram (right side view)

2.2.3 Pin Description

Pins	Name	Type	Driv.	Description
RF				
2 3	WF0_RFION_1 WF0_RFION_2	A		WF0 main path RF I/O
4 5	WF0_RFIOP_1 WF0_RFIOP_2	A		WF0 main path RF I/O
8 9	WF1_RFION_1 WF1_RFION_2	A		WF1 main path RF I/O
10 11	WF1_RFIOP_1 WF1_RFIOP_2	A		WF1 main path RF I/O
7	WF1_LNA_EXT	A		WF1 aux. path LNA input
1	WF0_LNA_EXT	A		WF0 aux. path LNA input
116	XTALIN	I		Crystal oscillator input
118	CLKOUTP	O		XO reference clock output
114	AVDD33_XTAL	P		3.3V XTAL Power Supply Pin
115 117	AVS33_XTAL_1 AVS33_XTAL_2	G		3.3V XTAL Ground Pin
6	AVDD33_WF0_TX	P		3.3V RF Channel 0 Suppoly Power
12	AVDD33_WF1_TX	P		3.3V RF Channel 1 Suppoly Power
13	AVDD33_WF1_TRX	P		1.65V to 3.3V RF Channel 1 Suppoly Power
113	AVDD33_WF_RFDIG	P		1.65V to 3.3V RF DIG and AFE Suppoly Power
119	AVDD33_WF_SX	P		1.65V to 3.3V RF Supply Power
120	AVDD33_WF0_TRX	P		1.65V to 3.3V RF Channel 0 Suppoly Power
WLAN LED				
85	WLED_N	O	4 mA	WLAN Activity LED
UART0 Lite				
28	UART_TXD0	O, IPD	4 mA	UART0 Lite TXD
29	UART_RXD0	I		UART0 Lite RXD
UART1 Lite				
111	UART_TXD1	O, IPU	4 mA	UART1 Lite TXD
112	UART_RXD1	I		UART1 Lite RXD
I2S				
14	I2S_SDI	I/O	4 mA	I2S data input
15	I2S_SDO	O, IPD	4 mA	I2S data output
16	I2S_WS	O	4 mA	I2S word select
17	I2S_CLK	I/O	4 mA	I2S clock
I2C				
19	I2C_SD	I/O	4 mA	I2C Data

Pins	Name	Type	Driv.	Description
18	I2C_SCLK	I/O	4 mA	I2C Clock
SPI				
24	SPI_MISO	I/O	4 mA	SPI Master input/Slave output
25	SPI_MOSI	I/O, IPD	4 mA	SPI Master output/Slave input
-23	SPI_CLK	O, IPU	4 mA	SPI clock
26	SPI_CS0	O	4 mA	SPI chip select0
22	SPI_CS1	O, IPD	4 mA	SPI chip select1
GPIO				
27	GPIO0	I/O, IPD	4 mA	General Purpose I/O
5-Port EPHY				
84	EPHY_LED0 _N_JTDO	I/O	4 mA	10/100 PHY Port #0 activity LED, JTAG_TDO
83	EPHY_LED1 _N_JTDI	I/O	4 mA	10/100 PHY Port #1 activity LED, JTAG_TDI
82	EPHY_LED2 _N_JTMS	I/O	4 mA	10/100 PHY Port #2 activity LED, JTAG_TMS
81	EPHY_LED3 _N_JTCLK	I/O	4 mA	10/100 PHY Port #3 activity LED, JTAG_CLK
80	EPHY_LED4 _N_JTRST_N	I/O,	4 mA	10/100 PHY Port #4 activity LED, JTAG_TRST_N
35	EPHY_VRT	A		Connect to an external resistor to provide accurate bias current
31	MDI_RP_P0	A		10/100 PHY Port #0 RXN
32	MDI_RN_P0	A		10/100 PHY Port #0 RXP
33	MDI_TP_P0	A		10/100 PHY Port #0 TXN
34	MDI_TN_P0	A		10/100 PHY Port #0 TXP
37	MDI_TP_P1	A		10/100 PHY Port #1 RXN
39	MDI_TN_P1	A		10/100 PHY Port #1 RXP
40	MDI_RP_P1	A		10/100 PHY Port #1 TXN
41	MDI_RN_P1	A		10/100 PHY Port #1 TXP
42	MDI_RP_P2	A		10/100 PHY Port #2 RXN
43	MDI_RN_P2	A		10/100 PHY Port #2 RXP
44	MDI_TP_P2	A		10/100 PHY Port #2 TXN
45	MDI_TN_P2	A		10/100 PHY Port #2 TXP
46	MDI_TP_P3	A		10/100 PHY Port #3 RXN
47	MDI_TN_P3	A		10/100 PHY Port #3 RXP
48	MDI_RP_P3	A		10/100 PHY Port #3 TXN
49	MDI_RN_P3	A		10/100 PHY Port #3 TXP
51	MDI_RP_P4	A		10/100 PHY Port #4 RXN
52	MDI_RN_P4	A		10/100 PHY Port #4 RXP
53	MDI_TP_P4	A		10/100 PHY Port #4 TXN

Pins	Name	Type	Driv.	Description
54	MDI_TN_P4	A		10/100 PHY Port #4 TXP
30	AVDD33_TX_P0	P		3.3V Supply Power for P0
36	AVDD33_COM	P		3.3V Supply Power for EPHY COM
38 50	AVDD33_TX_P1234_1 AVDD33_TX_P1234_2	P		3.3V Supply Power for P1 ~ P4
Misc.				
106	REF_CLKO	O, IPD	4 mA	Reference Clock Ouptut
108	PORST_N	I		Power on reset
107	WDT_RST_N	O	4 mA	Watchdog Reset
USB PHY				
58	AVDD33_USB	P		3.3 V USB PHY analog power supply
57	USB _VRT	A		Connect to an external 5.1 kΩ resistor for band-gap reference circuit
60	USB_DM	I/O		USB Port0 data pin Data-
59	USB _DP	I/O		USB Port0 data pin Data+
PCIe PHY				
105	PERST_N	O, IPD	4mA	PCIe device reset
98	PCIE_IO_VSS	G		PCIe Ground pin
101	AVDD12_PCIE	P		1.2 V PCIE PHY digital power supply
104	AVDD33_PCIE	P		3.3 V USB PHY analog power supply
103	PCIE_CKP0	O		External reference clock output (positive)
102	PCIE_CKN0	O		External reference clock output (negative)
97	PCIE_TXP0	I/O		PCIe0 differential transmit TX -
96	PCIE_TXN0	I/O		PCIe0 differential transmit TX -
99	PCIE_RXP0	I/O		PCIe0 differential receiver RX -
100	PCIE_RXN0	I/O		PCIe0 differential receiver RX -
PMU				
89 90	LXBK_1 LXBK_2	O		Buck Switching node
93	VOUT_FB	A		Buck vout feedback pin
87 88	AVDD33_SMPS_1 AVDD33_SMPS_2	P		Buck 3.3V Supply power
91 92	AVSS33_SMPS_1 AVSS33_SMPS_2	G		Buck Gound pin
94	AVDD33_DDRLDO	P		DDRLDO 3.3V Supply power
95	DDRLDO	O		DDRLDO 1.8V/2.5V output voltage
Power				

Pins	Name	Type	Driv.	Description
21 86 110	SOC_IO_V33D_1 SOC_IO_V33D_2 SOC_IO_V33D_3	P		3.3 V digital I/O power supply
20 55 56 71 72 74 75 109	SOC_CO _V12D_1 SOC_CO _V12D_2 SOC_CO _V12D_3 SOC_CO _V12D_4 SOC_CO _V12D_5 SOC_CO _V12D_6 SOC_CO _V12D_7 SOC_CO _V12D_8	P		1.2 V digital core power supply
EPAD	GND	G		Ground pin
NC				
64 65 68 69 70	NC_1 NC_2 NC_3 NC_4 NC_5	NC		No connected
Total: 120 pins				

Note:
- IPD : Internal pull-down
- IPU : Internal pull-up
- I : Input
- O : Output
- IO : Bi-directional
- P : Power
- G : Ground
- NC : Not connected

2.3 **Pin Sharing Schemes**

Some pins are shared with GPIO to provide maximum flexibility for system designers. The MT7628 provides up to 41 GPIO pins. Users can configure GPIO1_MODE and GPIO2_MODE registers in the System Control block to specify the pin function, or they can use the registers specified below. For more information, see the Programmer's Guide. Unless specified explicitly, all the GPIO pins are in input mode after reset.

2.3.1 GPIO pin share scheme

I/O Pad Group	Normal Mode	GPIO Mode
UART1	UART_RXD1	GPIO#46
	UART_TXD1	GPIO#45
WLED_AN	WLED_N (7628AN)	GPIO#44
P0_LED_AN	EPHY_LED0_N_JTDO (7628AN)	GPIO#43
P1_LED_AN	EPHY_LED1_N_JTDI (7628AN)	GPIO#42

I/O Pad Group	Normal Mode	GPIO Mode
P2_LED_AN	EPHY_LED2_N_JTMS (7628AN)	GPIO#41
P3_LED_AN	EPHY_LED3_N_JTCLK (7628AN)	GPIO#40
P4_LED_AN	EPHY_LED4_N_JTRST_N (7628AN)	GPO#39
WDT	WDT_RST_N	GPO#38
REFCLK	REF_CLKO	GPIO#37
PERST	PERST_N	GPIO#36
WLED_KN	WLED_N (7628KN)	GPIO#35
P0_LED_KN	EPHY_LED0_N_JTDO (7628KN)	GPIO#34
P1_LED_KN	EPHY_LED1_N_JTDI (7628KN)	GPIO#33
P2_LED_KN	EPHY_LED2_N_JTMS (7628KN)	GPIO#32
P3_LED_KN	EPHY_LED3_N_JTCLK (7628KN)	GPIO#31
P4_LED_KN	EPHY_LED4_N_JTRST_N (7628KN)	GPIO#30
SD	MDI_TN_P4	GPIO#29
	MDI_TP_P4	GPIO#28
	MDI_RN_P4	GPIO#27
	MDI_RP_P4	GPIO#26
	MDI_RN_P3	GPIO#25
	MDI_RP_P3	GPIO#24
	MDI_TN_P3	GPIO#23
	MDI_TP_P3	GPIO#22
UART2	MDI_TN_P2	GPIO#21
	MDI_TP_P2	GPIO#20
PWM1	MDI_RN_P2	GPO#19
PWM0	MDI_RP_P2	GPO#18
SPIS	MDI_RN_P1	GPIO#17
	MDI_RP_P1	GPIO#16
	MDI_TN_P1	GPO#15
	MDI_TP_P1	GPIO#14
UART0	UART_RXD0	GPIO#13
	UART_TXD0	GPIO#12
GPIO	GPIO0	GPIO#11
SPI	SPI_CS0	GPIO#10
	SPI_MISO	GPIO#9
	SPI_MOSI	GPIO#8
	SPI_CLK	GPIO#7
SPI_CS1	SPI_CS1	GPIO#6
I2C	I2C_SD	GPO#5

I/O Pad Group	Normal Mode	GPIO Mode
	I2C_SCLK	GPO#4
I2S	I2S_CLK	GPIO#3
	I2S_WS	GPIO#2
	I2S_SDO	GPIO#1
	I2S_SDI	GPO#0

2.3.2 UART1 pin share scheme

Controlled by the UART1_MODE register.

Pin Name	2'b00 UART-Lite #1	2'b01 GPIO	2'b10 PWM	2'b11 TRX_SW
UART1_RXD	UART1_RXD	GPIO#46	PWM_CH1	
UART1_TXD	UART1_TXD	GPIO#45	PWM_CH0	

2.3.3 MT7628AN EPHY LED pin share scheme

Controlled by the P#_LED_AN_MODE registers

Pin Name	Bootstrapping (DBG_JTAG_MODE=1)	Bootstrapping (DBG_JTAG_MODE=0)	
		P4_LED_AN_MODE =2'b00	P4_LED_AN_MODE =2'b01
EPHY_LED4_N_JTRST_N	JTAG_RST_N	EPHY_LED4_N	GPIO#39
		P3_LED_AN_MODE =2'b00	P3_LED_AN_MODE =2'b01
EPHY_LED3_N_JTCLK	JTAG_CLK	EPHY_LED3_N	GPIO#40
		P2_LED_AN_MODE =2'b00	P2_LED_AN_MODE =2'b01
EPHY_LED2_N_JTMS	JTAG_TMS	EPHY_LED2_N	GPIO#41
		P1_LED_AN_MODE =2'b00	P1_LED_AN_MODE =2'b01
EPHY_LED1_N_JTDI	JTAG_TDI	EPHY_LED1_N	GPIO#42
		P0_LED_AN_MODE =2'b00	P0_LED_AN_MODE =2'b01
EPHY_LED0_N_JTDO	JTAG_TDO	EPHY_LED0_N	GPIO#43

2.3.4 MT7628AN WLAN LED pin share scheme

Controlled by the WLED_AN_MODE registers

Pin Name	2'b00	2'b01
WLED_N	WLED_N	GPIO#44

2.3.5 MT7628KN EPHY LED pin share scheme

Controlled by the P#_LED_KN_MODE registers

Pin Name	Bootstrapping (DBG_JTAG_MODE=1)	Bootstrapping (DBG_JTAG_MODE=0)	
		P4_LED_KN_MODE =2'b00	P4_LED_KN_MODE =2'b01

Pin Name	Bootstrapping (DBG_JTAG_MODE=1)	Bootstrapping (DBG_JTAG_MODE=0)	
		P4_LED_KN_MODE =2'b00	P4_LED_KN_MODE =2'b01
EPHY_LED4_N_JTRST_N	JTAG_RST_N	EPHY_LED4_N	GPIO#30
		P3_LED_KN_MODE =2'b00	P3_LED_KN_MODE =2'b01
EPHY_LED3_N_JTCLK	JTAG_CLK	EPHY_LED3_N	GPIO#31
		P2_LED_KN_MODE =2'b00	P2_LED_KN_MODE =2'b01
EPHY_LED2_N_JTMS	JTAG_TMS	EPHY_LED2_N	GPIO#32
		P1_LED_KN_MODE =2'b00	P1_LED_KN_MODE =2'b01
EPHY_LED1_N_JTDI	JTAG_TDI	EPHY_LED1_N	GPIO#33
		P0_LED_KN_MODE =2'b00	P0_LED_KN_MODE =2'b01
EPHY_LED0_N_JTDO	JTAG_TDO	EPHY_LED0_N	GPIO#34

2.3.6 MT7628KN WLAN LED pin share scheme

Controlled by the WLED_KN_MODE registers

Pin Name	2'b00	2'b01
WLED_N	WLED_N	GPIO#35

2.3.7 PERST_N pin share scheme

Controlled by the PERST_ MODE register.

Pin Name	1'b0	1'b1
PERST_N	PERST_N	GPIO#36

2.3.8 WDT_RST_N pin share scheme

Controlled by the WDT _MODE register.

Pin Name	1'b0	1'b1
WDT_RST_N	WDT_RST_N	GPIO#37

2.3.9 REF_CLKO pin share scheme

Controlled by the REFCLK _MODE register.

Pin Name	1'b0	1'b1
REF_CLKO	REF_CLKO	GPIO#38

2.3.10 UART0 pin share scheme

Controlled by the UART0 _MODE register.

Pin Name	1'b0	1'b1
UART_TXD0	UART_TXD0	GPIO#12
UART_TXD0	UART_RXD0	GPIO#13

2.3.11 GPIO0 pin share scheme

Controlled by GPIO_MODE register.

Pin Name	2'b00	2'b01	2'b10	2'b11
GPIO0	GPIO#11	GPIO#11	REF_CLKO	PERST_N

2.3.12 SPI pin share scheme

Controlled by SPI_ MODE register.

Pin Name	1'b0	1'b1
SPI_CLK	SPI_CLK	GPO#7
SPI_MOSI	SPI_MOSI	GPO#8
SPI_MISO	SPI_MISO	GPIO#9
SPI_CS0	SPI_CS0	GPIO#10

2.3.13 SPI_CS1 pin share scheme

Controlled by SPI_CS1_MODE register.

Pin Name	2'b00	2'b01	2'b10
SPI_CS1	SPI_CS1	GPIO#6	REF_CLKO

2.3.14 I2C pin share scheme

Controlled by I2C_MODE register.

Pin Name	2'b00	2'b01
I2C_SCLK	I2C_SCLK	GPIO#4
I2C_SD	I2C_SD	GPIO#5

2.3.15 I2S pin share scheme

Controlled by I2S_MODE register.

Pin Name	2'b00	2'b01	2'b10
I2S_SDI	I2C_SCLK	GPIO#0	PCMDRX
I2S_SDO	I2C_SD	GPIO#1	PCMDTX
I2S_WS	I2C_SCLK	GPIO#2	PCMCLK
I2S_CLK	I2C_SD	GPIO#3	PCMFS

2.3.16 SD pin share scheme

Controlled by the EPHY_APGIO_AIO_EN[4:1] and SD_MODE registers

	EPHY_APGIO_AIO_EN[4:1] =4'b0000	EPHY_APGIO_AIO_EN[4:1] =4'b1111	
Pin Name		SD_MODE =2'b00	SD_MODE =2'b01
MDI_TP_P3	MDI_TP_P3	SD_WP	GPIO#22
MDI_TN_P3	MDI_TN_P3	SD_CD	GPIO#23
MDI_RP_P3	MDI_RP_P3	SD_D1	GPIO#24
MDI_RN_P3	MDI_RN_P3	SD_D0	GPIO#25
MDI_RP_P4	MDI_RP_P4	SD_CLK	GPIO#26
MDI_TN_P4	MDI_TN_P4	SD_D2	GPIO#27
MDI_RN_P4	MDI_RN_P4	SD_CMD	GPIO#28
MDI_TP_P4	MDI_TP_P4	SD_D3	GPIO#29

2.3.17 UART2 pin share scheme

Controlled by the EPHY_APGIO_AIO_EN[4:1] and UART2_MODE registers

	4'b0000	4'b1111			
Pin Name		2'b00	2'b01	2'b10	2'b11
MDI_TP_P2	MDI_TP_P2	UART_TXD2	GPIO#20	PWM_CH2	SD_D5
MDI_TN_P2	MDI_TN_P2	UART_RXD2	GPIO#21	PWM_CH3	SD_D4

2.3.18 PWM_CH0 pin share scheme

Controlled by the EPHY_APGIO_AIO_EN[4:1] and PWM0_MODE registers

	4'b0000	4'b1111			
Pin Name		2'b00	2'b01	2'b10	2'b11
MDI_RP_P2	MDI_RP_P2	PWM_CH0	GPIO#18		SD_D7

2.3.19 PWM_CH1 pin share scheme

Controlled by the EPHY_APGIO_AIO_EN[4:1] and PWM1_MODE registers

	4'b0000	4'b1111			
Pin Name		2'b00	2'b01	2'b10	2'b11
MDI_RN_P2	MDI_RN_P2	PWM_CH1	GPIO#19		SD_D6

2.3.20 SPIS pin share scheme

Controlled by the EPHY_APGIO_AIO_EN[4:1] and SPIS_MODE registers

	4'b0000	4'b1111			
Pin Name		2'b00	2'b01	2'b10	2'b11

	4’b0000	4’b1111			
Pin Name		2’b00	2’b01	2’b10	2’b11
MDI_TP_P1	MDI_TP_P1	SPIS_CS	GPIO#14		PWM_CH0
MDI_TN_P1	MDI_TN_P1	SPIS_CLK	GPIO#15		PWM_CH1
MDI_RP_P1	MDI_RP_P1	SPIS_MISO	GPIO#16		UART_TXD2
MDI_RN_P1	MDI_RN_P1	SPIS_MOSI	GPIO#17		UART_RXD2

2.3.21 Pin share function description

Pin Share Name	I/O	Pin Share Function description
PCMDTX	O	PCM Data Transmit DATA signal sent from the PCM host to the external codec.
PCMDRX	I	PCM Data Receive DATA signal sent from the external codec to the PCM host.
PCMCLK	I/O	PCM Clock The clock signal can be generated by the PCM host (Output direction), or provided by an external clock (input direction). The clock frequency should match the slot configuration of the PCM host. e.g. 4 slots, PCM clock out/in should be 256 kHz. 8 slots, PCM clock out/in should be 512 kHz. 16 slots, PCM clock out/in should be 1.024 MHz. 32 slots, PCM clock out/in should be 2.048 MHz. 64 slots, PCM clock out/in should be 4.096 MHz. 128 slots, PCM clock out/in should be 8.192 MHz.
PCMFS	I/O	PCM SYNC signal. In our design, the direction of this signal is independent of the direction of PCMCLK. Its direction and mode is configurable.
PWM_CH0	O	Pulse Width Modulation Channle 0
PWM_CH1	O	Pulse Width Modulation Channle 1
PWM_CH2	O	Pulse Width Modulation Channle 2
PWM_CH3	O	Pulse Width Modulation Channle 3

2.4 **Bootstrapping Pins Description**

Pin Name	Boot Strapping Signal Name	Description
UART_TXD1	DBG_JTAG_MODE	0: JTAG_MODE 1: EPHY_LED (default)
PERST_N	XTAL_FREQ_SEL	0: 25 MHz DIP 1: 40 MHz SMD

Pin Name	Boot Strapping Signal Name	Description
I2S_SDO	DRAM_TYPE	1: DDR1 0: DDR2 [note] This pin is valid for MT7628AN only. It needs to be pull-low for 7628KN which only supports DDR1.
{SPI_MOSI SPI_CLK, SPI_CS1}	CHIP_MODE[2:0]	A vector to set chip function/test/debug modes. 000: Boot from PLL (boot from SPI 3-Byte Addr) 001: Boot from PLL (boot from SPI 4-Byte Addr) 010: Boot from XTAL (boot from SPI 3-Byte Addr) 011: Boot from XTAL (boot from SPI 4-Byte Addr)
UART_TXD0	EXT_BGCK	1: Test Mode 0: Normal (default)

3. Maximum Ratings and Operating Conditions

3.1 **Absolute Maximum Ratings**

I/O supply voltage	3.63 V
Input, Output, or I/O Voltage	GND -0.3 V to Vcc +0.3 V

Table 3-1 Absolute Maximum Ratings

3.2 **Maximum Temperatures**

Maximum Junction Temperature (Plastic Package)	125 °C
Maximum Lead Temperature (Soldering 10 s)	260 °C

Table 3-2 Maximum Temperatures

3.3 **Operating Conditions**

I/O supply voltage	3.3 V +/- 10%
DDR1 supply voltage	2.5 V +/- 5%
DDR2 supply voltage	1.8 V +/- 5%
Core supply voltage	1.2 V +/- 10%
Ambient Temperature Range	-20 to 55 °C

Table 3-3 Operating Conditions

Table 3-4 Thermal Characteristics

3.4 **Storage Conditions**

The calculated shelf life in a sealed bag is 12 months if stored between 0 °C and 40 °C at less than 90% relative humidity (RH). After the bag is opened, devices that are subjected to solder reflow or other high temperature processes must be handled in the following manner:

- Mounted within 168 hours of factory conditions, i.e. < 30 **°**C at 60% RH.
- Storage humidity needs to maintained at < 10% RH.
- Baking is necessary if the customer exposes the component to air for over 168 hrs, baking conditions: 125 °C for 8 hrs.

3.5 **External Xtal Specfication**

Frequency	25 MHz/ 40 Mhz
Frequency offset	+/-7ppm @ 25 ℃ +/-15ppm @ -40~85 ℃
Load Capacitance (CL)	13pF
Shunt Capacitance (Co)	7.0 pF MAX

Pulling Sensitivity (TS)	20ppm /pF (Load @ 13pF)

Table 3-5 External Xtal Specifications

3.6 DC Electrical Characteristics

MT7628A (2T2R(HT40/MCS15), LAN x 4,WANx1, LAN to WAN, USB (SAMBA), PCIe OFF)						
Parameters	Sym	Conditions	Min	Typ	Max	Unit
3.3 V supply voltage (IO)	Vddc33		2.97	3.3	3.63	V
2.5V supply voltage (DDR1)	Vdd25		2.375	2.5	2.625	V
1.8 V supply voltage (DDR2)	Vdd18		1.71	1.8	1.89	V
1.2 V supply voltage	Vdd12		1.14	1.2	1.32	V
3.3 V current consumption	Icc33			440	1000	mA
1.2 V current consumption	Icc12			150	380	mA
1.8V DDR2 Current	Icc18			50	170	mA

MT7628K (2T2R(HT40/MCS15), LAN x 4, WANx1, w/o USB, w/o PCIe)						
Parameters	Sym	Conditions	Min	Typ	Max	Unit
3.3 V supply voltage (IO)	Vddc33		2.97	3.3	3.63	V
2.5V supply voltage (DDR1)	Vdd25		2.375	2.5	2.625	V
1.8 V supply voltage (DDR2)	Vdd18		1.71	1.8	1.89	V
1.2 V supply voltage	Vdd12		1.14	1.2	1.32	V
3.3 V current consumption	Icc33			380	850	mA
1.2 V current consumption	Icc12			130	380	mA
1.8V DDR2 Current	Icc18			50	100	mA

Table 3-6 DC Electrical Characteristics

Vdd=2.5V (DDR1)	Min	Typ	Max
Vdd	2.375	2.5	2.625
VIH	VREF+0.15		Vdd25+0.3
VIL	-0.3		VREF-0.15
VOH	0.8*Vdd25		

VOL			0.2*Vdd25
IOL			
IOH			

Table 3-7 Vdd 2.5V Electrical Characteristics

Vdd=1.8V (DDR2)	Min	Typ	Max
Vdd	1.71	1.8	1.89
VIH	VREF+0.125		Vdd18+0.3
VIL	-0.3		VREF-0.125
VOH	1.42		
VOL			0.28
IOL			
IOH			

Table 3-8 Vdd 1.8V Electrical Characteristics

Vdd=3.3V	Min	Typ	Max
Vdd	2.97V	3.3V	3.63V
VIH	2.0V		Vdd33+0.3
VIL	-0.3		0.8V
VOH	2.4V		
VOL			0.4V
IOL			
IOH			

Table 3-9 Vdd 3.3V Electrical Characteristics

3.7 **AC Electrical Characteristics**

3.7.1 DDR2 SDRAM Interface

The DDR2 SDRAM interface complies with 200 MHz timing requirements for standard DDR2 SDRAM. The interface drivers are SSTL_18 drivers matching the EIA/JEDEC standard JESD8-15A.

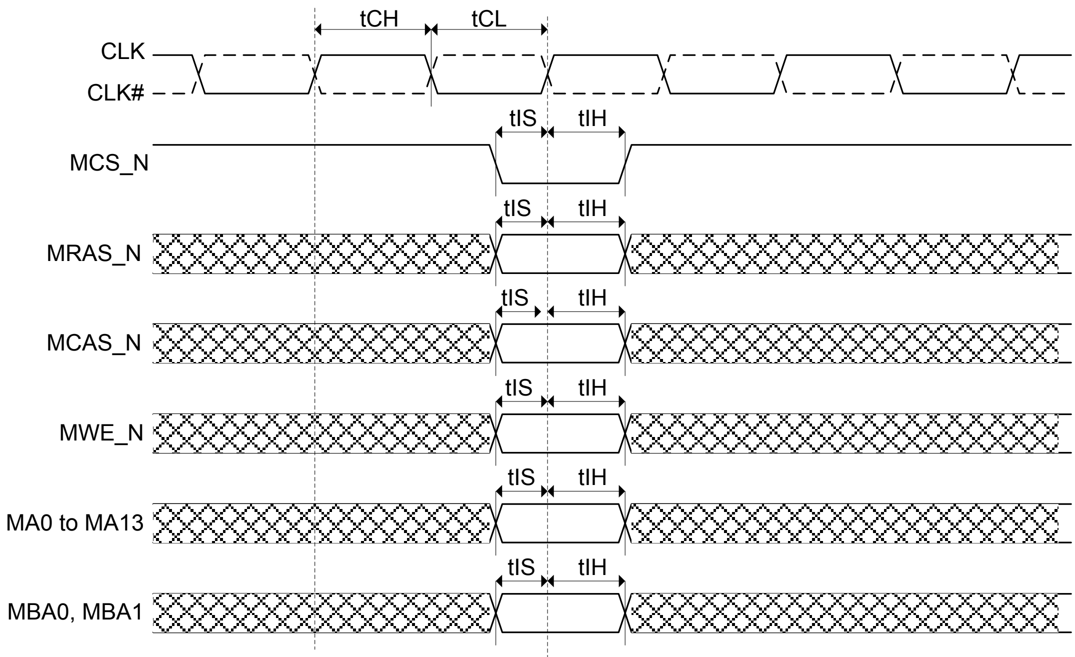


Figure 3-1 DDR2 SDRAM Command

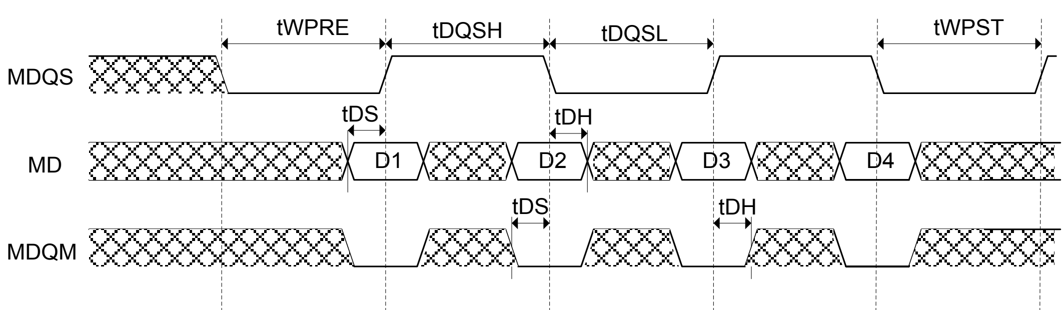


Figure 3-2 DDR2 SDRAM Write data

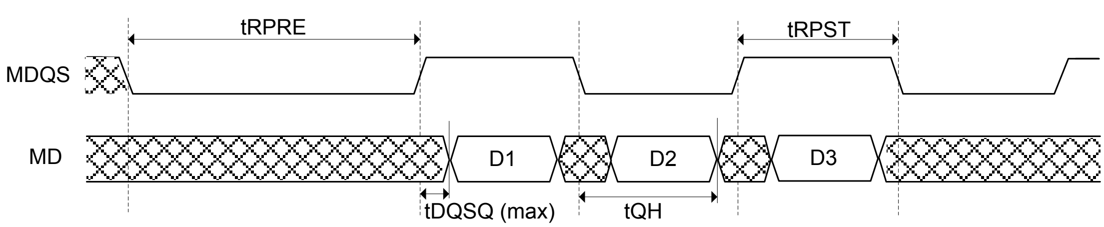


Figure 3-3 DDR2 SDRAM Read data

Symbol	Description	Min	Max	Unit	Remark
tCK(avg)	Clock cycle time	5	-	ns	
tAC	DQ output access time from SDRAM CLK	-0.6	0.6	ns	
tDQSCK	DQS output access time from SDRAM CLK	-0.5	0.5	ns	
tCH	SDRAM CLK high pulse width	0.48	0.52	tCK(avg)	
tCL	SDRAM CLK low pulse width	0.48	0.52	tCK(avg)	
tHP	SDRAM CLK half period	Min(tCH,tCL)	-	ns	
tIS	Address and control input setup time	0.75	-	ns	
tIH	Address and control input hold time	0.75	-	ns	
tDQSQ	Data skew of DQS and associated DQ	-	0.4	ns	
tQH	DQ/DQS output hold time from DQS	tHP-0.5	-	ns	
tRPRE	DQS read preamble	0.9	1.1	tCK	
tRPST	DQS read postamble	0.4	0.6	tCK	
tDQSS	DQS rising edge to CK rising edge	-0.25	0.25	tCK	
tDQSH	DQS input-high pulse width	0.35	-	tCK	
tDQSL	DQS input-low pulse width	0.35	-	tCK	
tDSS	DQS falling edge to SDRAM CLK setup time	0.2	-	tCK	
tDSH	DQS falling edge hold time from SDRAM CLK	0.2	-	tCK	
tWPRE	DQS write preamble	0.35	-	tCK	
tWPST	DQS write postamble	0.4	0.6	tCK	
tDS	DQ and DQM input setup time	*0.4	-	ns	
tDH	DQ and DQM input hold time	*0.4	-	ns	

Table 3-10 DDR2 SDRAM Interface Diagram Key

NOTE: Depends on slew rate of DQS and DQ/DQM for single ended DQS.

3.7.2 SPI Interface

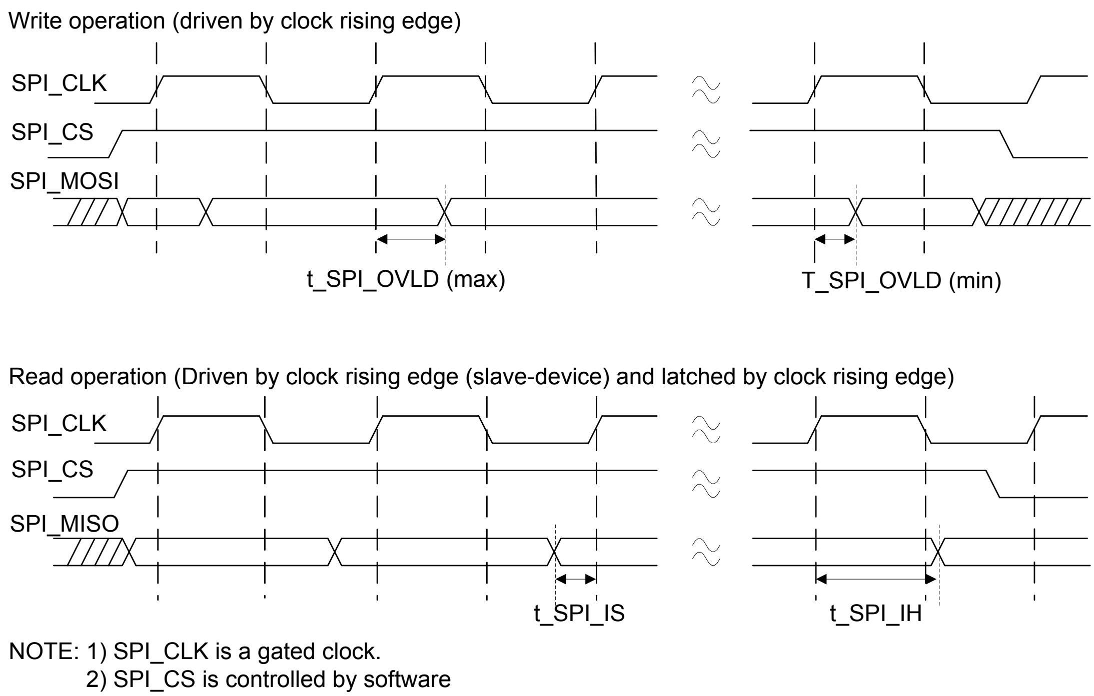


Figure 3-4 SPI Interface

Symbol	Description	Min	Max	Unit	Remark
t_SPI_IS	Setup time for SPI input	6.0	-	ns	
t_SPI_IH	Hold time for SPI input	-1.0	-	ns	
t_SPI_OVLD	SPI_CLK to SPI output valid	-2.0	3.0	ns	output load: 5 pF

Table 3-11 SPI Interface Diagram Key

3.7.3 I²S Interface

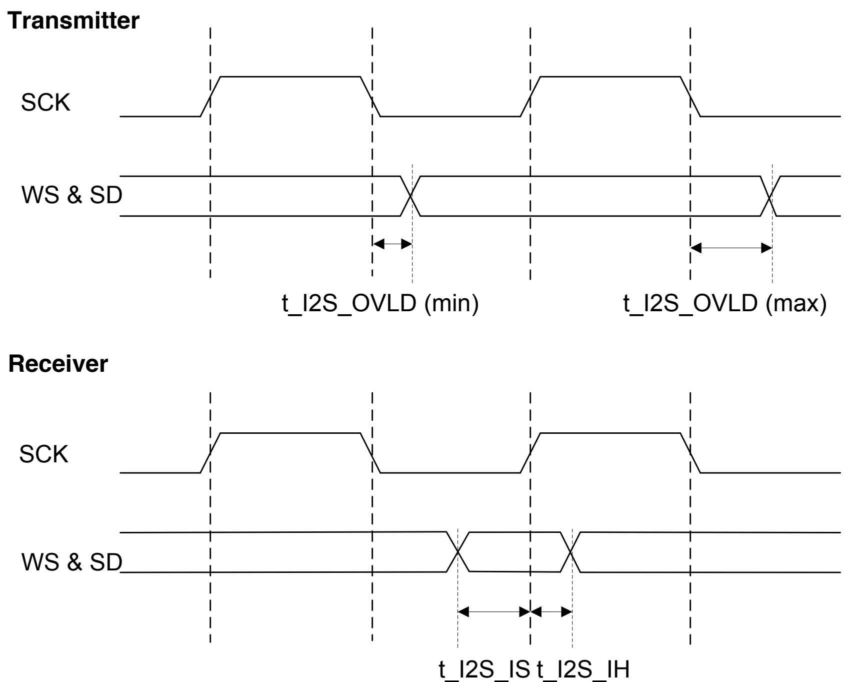


Figure-3-5 I2S Interface

Symbol	Description	Min	Max	Unit	Remark
t_I2S_IS	Setup time for I2S input (data & WS)	3.5	-	ns	
t_I2S_IH	Hold time for I2S input (data & WS)	0.5	-	ns	
t_I2S_OVLD	I2S_CLK to I2S output (data & WS) valid	2.5	10.0	ns	output load: 5 pF

Table 3-12 I2S Interface Diagram Key

3.7.4 PCM Interface

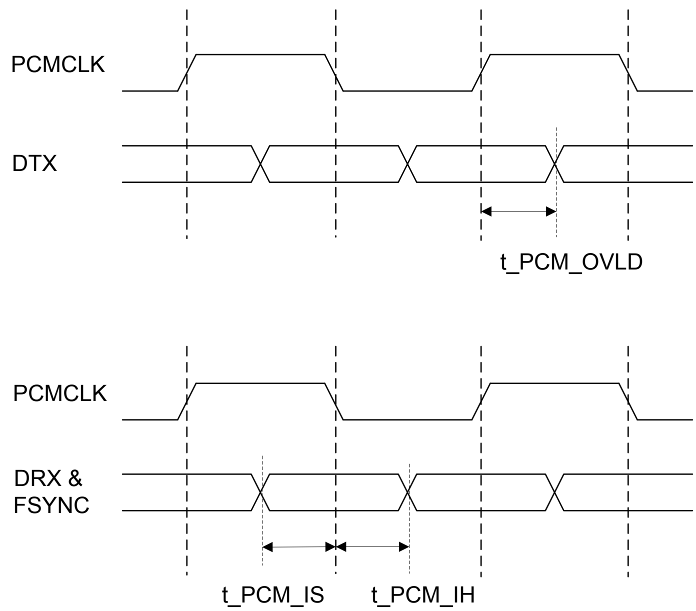
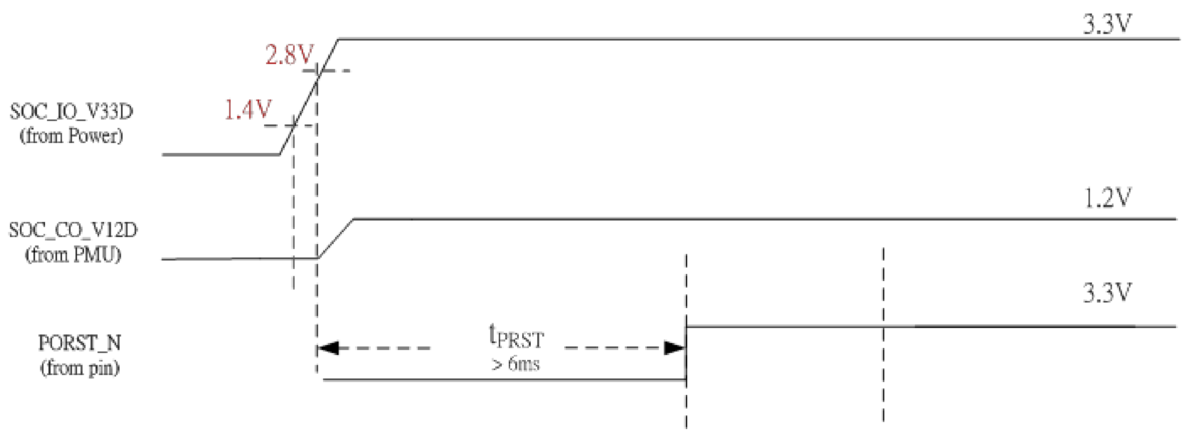


Figure 3-6 PCM Interface

Symbol	Description	Min	Max	Unit	Remark
t_PCM_IS	Setup time for PCM input to PCM_CLK fall	3.0	-	ns	
t_PCM_IH	Hold time for PCM input to PCM_CLK fall	1.0	-	ns	
t_PCM_OVLD	PCM_CLK rise to PCM output valid	10.0	35.0	ns	output load: 5 pF

Table 3-13 PCM Interface Diagram Key

3.7.5 Power On Sequence



Symbol	Description	Min	Max	Unit
t_{PRST}	External Power-on Reset Period	6.0		ms

Figure 3-7 Power ON Sequence

Table 3-14 Power ON Sequence Diagram Key

3.8 **Package Physical Dimensions**

3.8.1 DR-QFN (10 mm x 10 mm) 128 pins

3.8.1.1 Top View

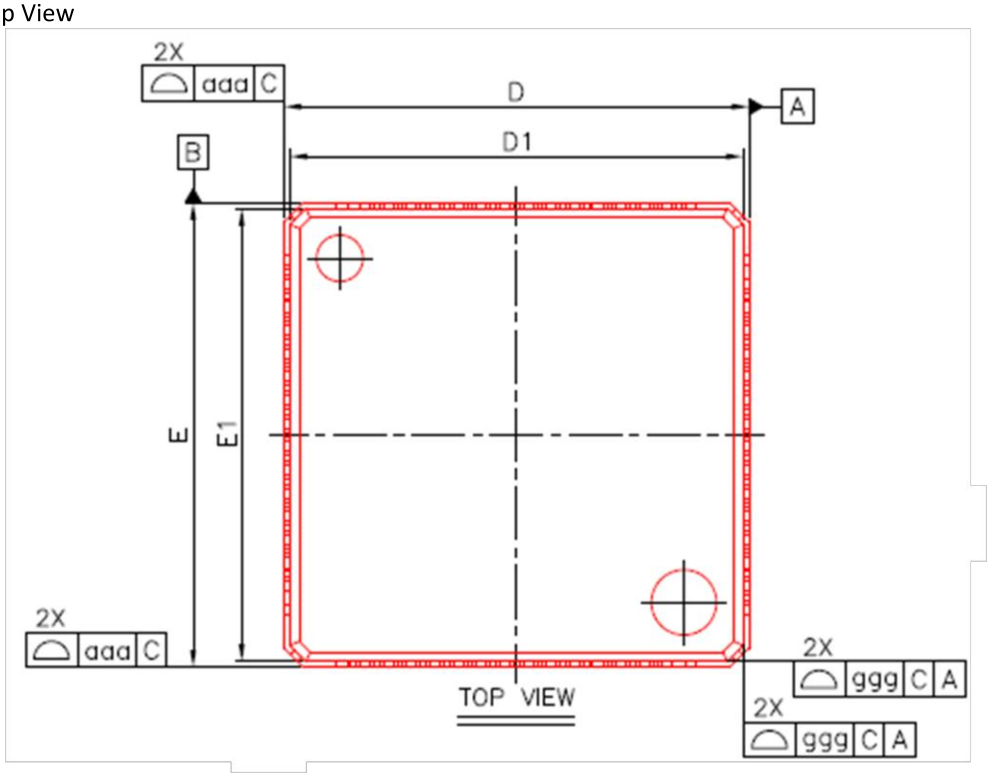
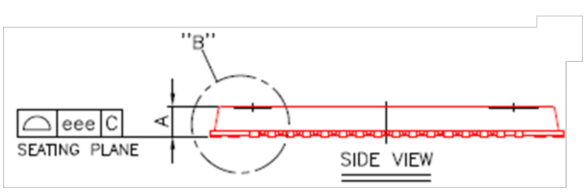


Figure 3-8 Top View

3.8.1.2 Side View



3.8.1.3 “B” Expanded

Figure 3-9 Side View

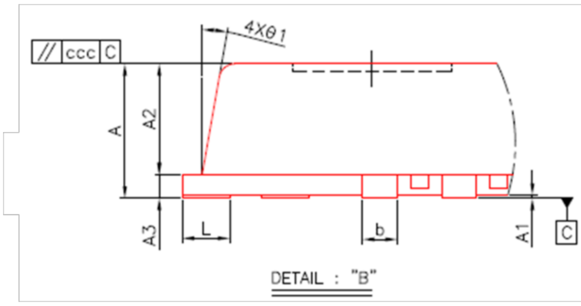


Figure 3-10 "B" Expanded

3.8.1.4 Bottom View

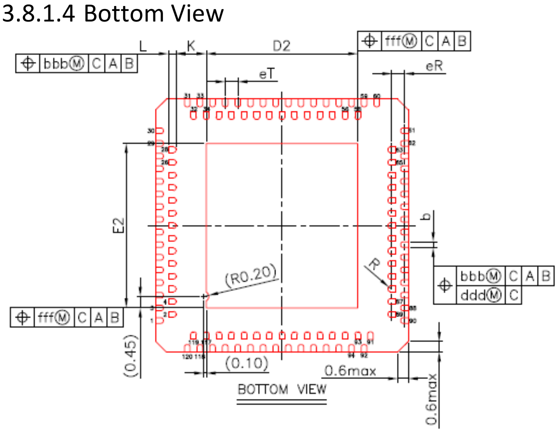


Figure 3-11 Botton view

3.8.1.5 Package Diagram Key

Item	SYMBOL	MIN.	NOM.	MAX.
TOTAL THICKNESS	A	0.80	0.85	0.90
LEAD STAND OFF.	A1	0.00	0.02	0.05
MOLD THICKNESS	A2	0.65	0.70	0.75
L/F THICKNESS	A3	0.15 REF.		
LEAD WIDTH	b	0.18	0.22	0.30
PACKAGE SIZE	D	9.90	10.00	10.10
	E			
Mold Edge size	D1	9.75 BSC		
	E1	9.75 BSC		
E-PAD size	D2	5.90	6.00	6.10
	E2	6.40	6.50	6.60
LEAD LENGTH	L	0.20	0.30	0.40
LEAD PITCH (BSC.)	eT	0.50 BSC		
LEAD PITCH (BSC.)	eR	0.50 BSC		
ANGLE	θ1	5°	---	15°
LEAD ARC	R	0.09	---	0.14
Lead to E-PAD Toler-ance	K	0.20	---	---
PKG EDGE TOLER-ANCE	aaa	0.10		
PACKAGE PROFILE OF A SURFACE	bbb	0.10		
LEAD PROFILE OF A SURFACE	ccc	0.10		
LEAD POSITION	ddd	0.05		
LEAD PROFILE OF A SURFACE	eee	0.08		
EPAD POSTION	fff	0.10		
Mold edge OF A & C SURFACE	ggg	0.20		

3.8.2 DR-QFN (12 mm x 12 mm) 156 pins

3.8.2.1 Top View

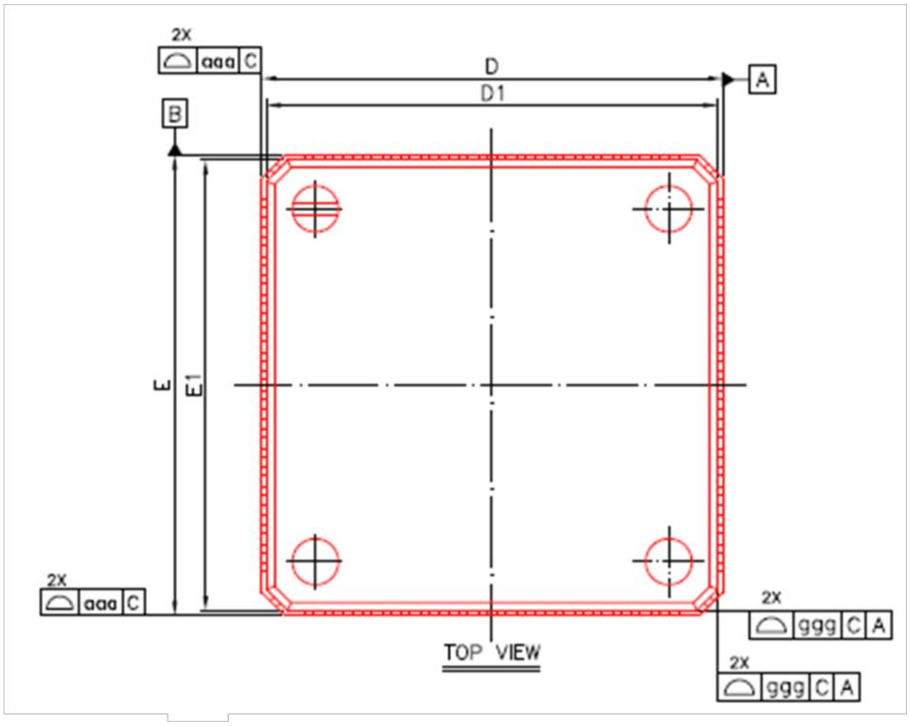


Figure 3-12 Top View

3.8.2.2 Side View

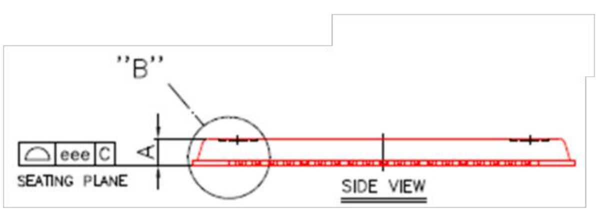


Figure 3-13 Side View

3.8.2.3 "B" Expanded

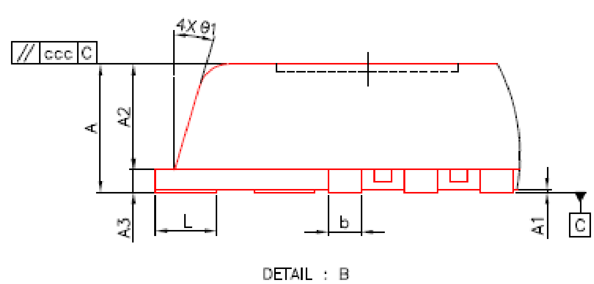


Figure 3-14 "B" Expanded

3.8.2.4 Bottom View

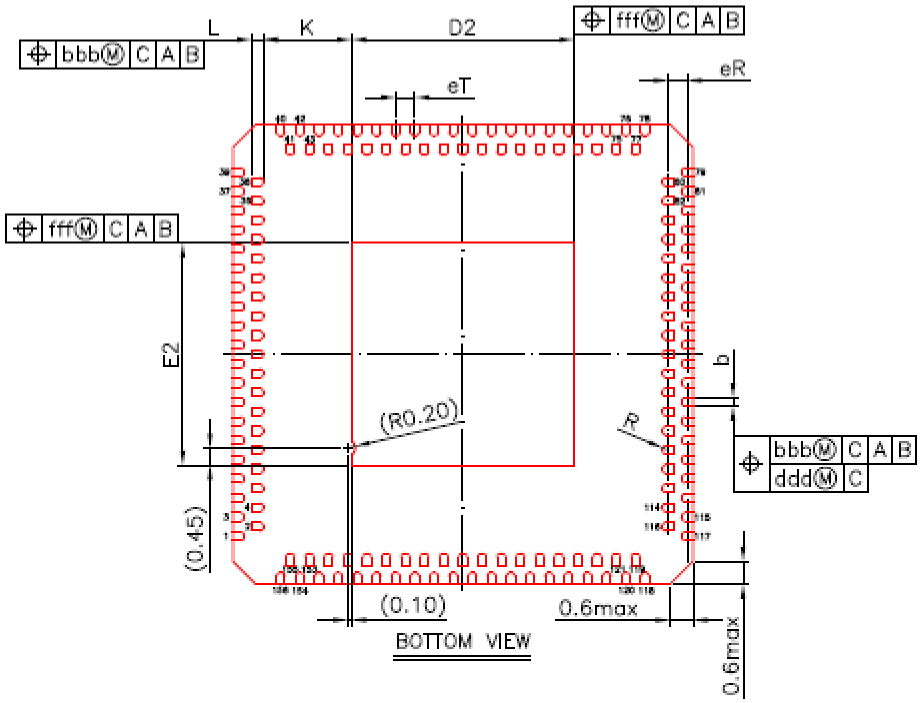


Figure 3-15 Bottom View

3.8.2.5 Package Diagram Key

Item	SYMBOL	MIN.	NOM.	MAX.
TOTAL THICKNESS	A	0.80	0.85	0.90
LEAD STAND OFF.	A1	0.00	0.02	0.05
MOLD THICKNESS	A2	0.65	0.70	0.75
L/F THICKNESS	A3	0.15 REF.		
LEAD WIDTH	b	0.18	0.22	0.30
PACKAGE SIZE	D	11.90	12.00	12.10
	E			
Mold Edge size	D1	11.75 BSC		
	E1	11.75 BSC		
E-PAD size	D2	5.70	5.80	5.90
	E2	5.70	5.80	5.90
LEAD LENGTH	L	0.20	0.30	0.40
LEAD PITCH (BSC.)	eT	0.50 BSC		
LEAD PITCH (BSC.)	eR	0.50 BSC		
ANGLE	θ1	5°	---	15°
LEAD ARC	R	0.09	---	0.14
Lead to E-PAD Toler-ance	K	0.20	---	---
PKG EDGE TOLER-ANCE	aaa	0.10		
PACKAGE PROFILE OF A SURFACE	bbb	0.10		
LEAD PROFILE OF A SURFACE	ccc	0.10		
LEAD POSITION	ddd	0.05		
LEAD PROFILE OF A SURFACE	eee	0.08		
EPAD POSTION	fff	0.10		
Mold edge OF A & C SURFACE	ggg	0.20		

3.8.3 MT7628 AN/KN marking



YYWW: Date code
LLLLLLLLL : Lot number
“.” : Pin #1 dot

Figure 3-16 MT7620AN top marking



YYWW: Date code
LLLLLLLLLL : Lot number
“.” : Pin #1 dot

Figure 3-17 MT7628KN top marking

3.8.4 Reflow profile guideline

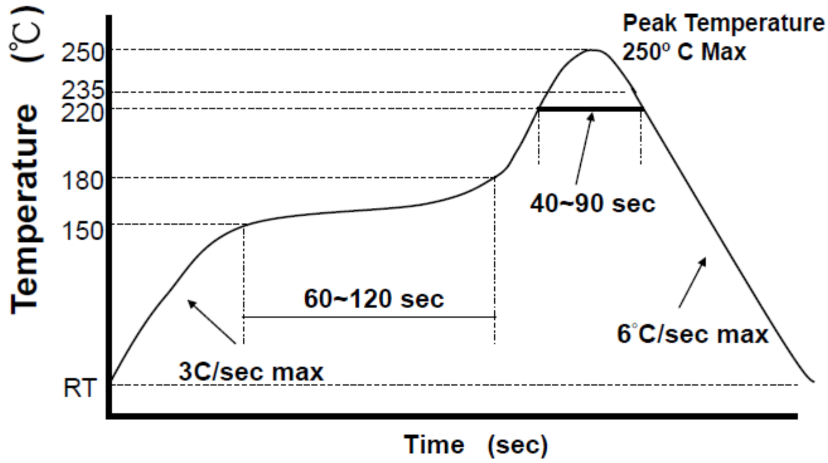


Figure 3-18 Reflow profile for MT7628

Notes;
1. Reflow profile guideline is designed for SnAgCulead-free solder paste.
2. Reflow temperature is defined at the solder ball of package/or the lead of package.
3. MTK would recommend customer following the solder paste vendor's guideline to design a profile appropriate your line and products.
4. Appropriate N2 atmosphere is recommended since it would widen the process window and mitigate the risk for having solder open issues.

4. Abbreviations

Abbrev.	Description
AC	Access Category
ACK	Acknowledge/ Acknowledgement
ACPR	Adjacent Channel Power Ratio
AD/DA	Analog to Digital/Digital to Analog converter
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AGC	Auto Gain Control
AIFS	Arbitration Inter-Frame Space
AIFSN	Arbitration Inter-Frame Spacing Number
ALC	Asynchronous Layered Coding
A-MPDU	Aggregate MAC Protocol Data Unit
A-MSDU	Aggregation of MAC Service Data Units
AP	Access Point
ASIC	Application-Specific Integrated Circuit
ASME	American Society of Mechanical Engineers
ASYNC	Asynchronous
BA	Block Acknowledgement
BAC	Block Acknowledgement Control
BAR	Base Address Register
BBP	Baseband Processor
BGSEL	Band Gap Select
BIST	Built-In Self-Test
BSC	Basic Spacing between Centers
BJT	
BSSID	Basic Service Set Identifier
BW	Bandwidth
CCA	Clear Channel Assessment
CCK	Complementary Code Keying
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CCX	Cisco Compatible Extensions
CF-END	Control Frame End
CF-ACK	Control Frame Acknowledgement
CLK	Clock

Abbrev.	Description
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSR	Control Status Register
CTS	Clear to Send
CW	Contention Window
CWmax	Maximum Contention Window
CWmin	Minimum Contention Window
DAC	Digital-To-Analog Converter
DCF	Distributed Coordination Function
DDONE	DMA Done
DDR	Double Data Rate
DFT	Discrete Fourier Transform
DIFS	DCF Inter-Frame Space
DMA	Direct Memory Access
DSP	Digital Signal Processor
DW	DWORD
EAP	Expert Antenna Processor
EDCA	Enhanced Distributed Channel Access
EECS	EEPROM chip select
EEDI	EEPROM data input
EEDO	EEPROM data output
EEPROM	Electrically Erasable Programmable Read-Only Memory
eFUSE	electrical Fuse
EESK	EEPROM source clock
EIFS	Extended Inter-Frame Space
EIV	Extend Initialization Vector
EVM	Error Vector Magnitude
FDS	Frequency Domain Spreading
FEM	Front-End Module
FEQ	Frequency Equalization
FIFO	First In First Out
FSM	Finite-State Machine
GF	Green Field
GND	Ground
GP	General Purpose
GPO	General Purpose Output
GPIO	General Purpose Input/Output

Abbrev.	Description
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HT	High Throughput
HTC	High Throughput Control
ICV	Integrity Check Value
IFS	Inter-Frame Space
iNIC	Intelligent Network Interface Card
IV	Initialization Vector
I^2C	Inter-Integrated Circuit
I^2S	Integrated Inter-Chip Sound
I/O	Input/Output
IPI	Idle Power Indicator
IQ	In phase/Quadrature phase
JEDEC	Joint Electron Devices Engineering Council
JTAG	Joint Test Action Group
kbps	kilo (1000) bits per second
KB	Kilo (1024) Bytes
LDO	Low-Dropout Regulator
LDODIG	LDO for DIGital part output voltage
LED	Light-Emitting Diode
LNA	Low Noise Amplifier
LO	Local Oscillator
L-SIG	Legacy Signal Field
MAC	Medium Access Control
MCU	Microcontroller Unit
MCS	Modulation and Coding Scheme
MDC	Management Data Clock
MDIO	Management Data Input/Output
MEM	Memory
MFB	MCS Feedback
MFS	MFB Sequence
MIC	Message Integrity Code
MIMO	Multiple-Input Multiple-Output
MLNA	Monolithic Low Noise Amplifier
MM	Mixed Mode
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MPDU	MAC Protocol Data Units
MSB	Most Significant Bit

Abbrev.	Description
NAV	Network Allocation Vector
NAS	Network-Attached Server
NAT	Network Address Translation
NDP	Null Data Packet
NVM	Non-Volatile Memory
ODT	On-die Termination
Oen	Output Enable
OFDM	Orthogonal Frequency-Division Multiplexing
OSC	Open Sound Control
PA	Power Amplifier
PAPE	Provider Authentication Policy Extension
PBC	Push Button Configuration
PBF	Packet Buffer
PCB	Printed Circuit Board
PCF	Point Coordination Function
PCM	Pulse-Code Modulation
PHY	Physical Layer
PIFS	PCF Interframe Space
PLCP	Physical Layer Convergence Protocol
PLL	Phase-Locked Loop
PME	Physical Medium Entities
PMU	Power Management Unit
PN	Packet Number
PROM	Programmable Read-Only Memory
PSDU	Physical layer Service Data Unit
PSI	Power supply Strength Indication
PSM	Power Save Mode
PTN	Packet Transport Network
QoS	Quality of Service
RDG	Reverse Direction Grant
RAM	Random Access Memory
RF	Radio Frequency
RGMII	Reduced Gigabit Media Independent Interface
RH	Relative Humidity
RoHS	Restriction on Hazardous Substances
ROM	Read-Only Memory

Abbrev.	Description
RSSI	Received Signal Strength Indication (Indicator)
RTS	Request to Send
RvMII	Reverse Media Independent Interface
Rx	Receive
RXD	Received Data
RXINFO	Receive Information
RXWI	Receive Wireless Information
S	Stream
SDXC	Secure Digital eXtended Capacity
SDIO	Secure Digital Input Output
SDRAM	Synchronous Dynamic Random Access Memory
SEC	Security
SGI	Short Guard Interval
SIFS	Short Inter-Frame Space
SoC	System-on-a-Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSCG	Spread Spectrum Clock Generator
STBC	Space–Time Block Code
SW	Switch Regulator
TA	Transmitter Address
TBTT	Target Beacon Transmission Time
TDLS	Tunnel Direct Link Setup
TKIP	Temporal Key Integrity Protocol
TRSW	Tx/Rx Switch
TSF	Timing Synchronization Function

Abbrev.	Description
TSSI	Transmit Signal Strength Indication
Tx	Transmit
TxBF	Transmit Beamforming
TXD	Transmitted Data
TXDAC	Transmit Digital-Analog Converter
TXINFO	Transmit Information
TXOP	Opportunity to Transmit
TXWI	Tx Wireless Information
UART	Universal Asynchronous Rx/ Tx
USB	Universal Serial Bus
UTIF	Universal Test Interface
VGA	Variable Gain Amplifier
VCO	Voltage Controlled Amplifier
VIH	High Level Input Voltage
VIL	Low Level Input Voltage
VoIP	Voice over IP
WCID	Wireless Client Identification
WEP	Wired Equivalent
WI	Wireless Information
WIV	Wireless Information Valid
WMM	Wi-Fi Multimedia
WPA	Wi-Fi Protected Access
WPDMA	Wireless Polarization Division Multiple Access
WS	Word Select

5. Revision History

Rev	Date	Description
1.0	2012/07/09	Initial Release
1.1	2012/07/18	Update SPI_WP/SPI_HOLD GPO table
1.2	2012/08/20	Fix DRQFN internal pad size typo
1.3	2012/09/12	Add IR reflow guideline

responsibility for any inaccuracies that may be contained in this document. Ralink reserves the right to make changes in its products to improve function, performance, reliability, and to attempt to supply the best product possible.

MT7628/7688
802.11 b/g/n Wi-Fi APSoC chip
EEPROM Content Programming guide

Version: 2.0
Release date: 2014-09-15

Document Revision History

EEPROM Changes LOG

Revision	Date	Author	Change Log
1.0	2014/7/16	PeterCT Wu	Formal Released
2.0	2014/9/15	PeterCT Wu	1. Define MAC0: 0x28~0x2D, MAC1: 0x2E ~ 0x33 2. Define 0x24[4] : 1x1 downgrade package 3. Define 0x24[3:0]: IO setting (for 7628N) 4. Re-define 0xF8[7:0] definition (delete bit[7]:valid) and default is 0xA 5. Define 0xC6 ~ 0xD6 (Temperature compensation) 00 00 00 1A 22 2A 31 35 - 01 35 39 40 46 4D 7F 7F - 7F 6. Define 0xF4=C0 (XTAL calibration) 7. Define TSSI offset (AN) 0x57=CA, 0x5D=CA (KN) 0x57=C8, 0x5D=C8 8. Remove 0x40 (define on 0x35) 9. Rev2_0

Table of Contents

1 General Description

1.1 General Descriptions

The MT7628/7688 EEPROM layout provides configuration for vendor/product ID, MAC Address, SW setting, RF TX power setting.

2 MT7628/7688 EEPROM Layout

Module name: EEPROM Base address: (+0h)

Name	Offset	Type	Byte	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CHIP_ID	0000	DC	1	76	CHIP_ID[15:8]							
			0	28	CHIP_ID[7:0]							
EEPROM_REV	0002	DC	1	02	REL_REV							
			0	00	ENG_REV							
WLAN_MAC_B1B0	0004	RW	1	0C	MAC_ADDR_L[15:8]							
			0	00	MAC_ADDR_L[7:0]							
WLAN_MAC_B3B2	0006	RW	1	E1	MAC_ADDR_M[15:8]							
			0	43	MAC_ADDR_M[7:0]							
WLAN_MAC_B5B4	0008	RW	1	28	MAC_ADDR_H[15:8]							
			0	76	MAC_ADDR_H[7:0]							
MAC0_B1B0	0028	RW	1	0C	MAC0_ADDR_L[15:8]							
			0	00	MAC0_ADDR_L[7:0]							
MAC0_B3B2	002A	RW	1	E1	MAC0_ADDR_M[15:8]							
			0	43	MAC0_ADDR_M[7:0]							
MAC0_B5B4	002C	RW	1	29	MAC0_ADDR_H[15:8]							
			0	76	MAC0_ADDR_H[7:0]							
MAC1_B1B0	002E	RW	1	0C	MAC1_ADDR_L[15:8]							
			0	00	MAC1_ADDR_L[7:0]							
MAC1_B3B2	0030	RW	1	E1	MAC1_ADDR_M[15:8]							
			0	43	MAC1_ADDR_M[7:0]							
MAC1_B5B4	0032	RW	1	2A	MAC1_ADDR_H[15:8]							
			0	76	MAC1_ADDR_H[7:0]							
NIC_CONFG_0	0034	RW	1	34	EXT_PA_ANTSEL		BOARD_TYPE			EXT_PA_DRV	EXT_2P4G_PA	EXT_5G_PA
			0	22	TX_PATH				RX_PATH			
NIC_CONFG_1	0036	RW	1	00	DAC_TEST		TSSI_COMP	ANT_DIV_CTRL				BW_40M_2P4G
			0	00	WPS				WF1_AUX	WF0_AUX	TX_POWER	HW_RADIO
COUNTRY_REG	39	RW	0	00	BAND_2P4G							
LED_MODE	3B	RW	0	01	LED_CTRL							
NIC_CONFG_2	0042	RW	1	00					TEMP_COMP	XTAL_OPT		ANT_DIV
			0	22	TX_STREAM				RX_STREAM			
EXT_LNA_GAIN	44	RW	0	00	EXT_LNA_2P4G							
RSSI_OFST	0046	RW	1	00	RSSI1_OFST							
			0	00	RSSI0_OFST							
TX_POWER_DELTA	50	RW	0	82	DELTA_EN	DELTA_IN	DELTA					

						C	
TEMP_SEN_CAL	55	**RW**	0	B0	TEMP_COMP_EN	THADC_SLOP	
TX0_PA_TSSI_LSB	56	**RW**	0	C0	TX0_PA_TSSI_OFST		TX0_PA_TSSI_SLOP
TX0_PA_TSSI_MSB	57	**RW**	0	CC	TX0_PA_TSSI_OFST		
TX0_POWER	58	**RW**	0	23	TX0_TX_PWR		
TX0_PWR_OFST_L	59	**RW**	0	00	TX0_TX_PWR_EN	TX0_TX_PWR_INC	TX0_TX_PWR_OFST_L
TX0_PWR_OFST_M	5A	**RW**	0	00	TX0_TX_PWR_EN	TX0_TX_PWR_INC	TX0_TX_PWR_OFST_M
TX0_PWR_OFST_H	5B	**RW**	0	00	TX0_TX_PWR_EN	TX0_TX_PWR_INC	TX0_TX_PWR_OFST_H
TX1_PA_TSSI_LSB	5C	**RW**	0	40	TX1_PA_TSSI_OFST		TX1_PA_TSSI_SLOP
TX1_PA_TSSI_MSB	5D	**RW**	0	CC	TX1_PA_TSSI_OFST		
TX1_POWER	5E	**RW**	0	23	TX1_TX_PWR		
TX1_PWR_OFST_L	5F	**RW**	0	00	TX1_TX_PWR_EN	TX1_TX_PWR_INC	TX1_TX_PWR_OFST_L
TX1_PWR_OFST_M	60	**RW**	0	00	TX1_TX_PWR_EN	TX1_TX_PWR_INC	TX1_TX_PWR_OFST_M
TX1_PWR_OFST_H	61	**RW**	0	00	TX1_TX_PWR_EN	TX1_TX_PWR_INC	TX1_TX_PWR_OFST_H
TX_PWR_CCK_0	A0	**RW**	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_CCK_1	A1	**RW**	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_OFDM_0	A2	**RW**	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_OFDM_1	A3	**RW**	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_OFDM_2	A4	**RW**	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_OFDM_3	A5	**RW**	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_OFDM_4	A6	**RW**	0	C6	TX_P	TX_P	TX_PWR_DELTA

					WR_COMP_EN	WR_INC	
TX_PWR_HT_MCS_0	A7	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_HT_MCS_1	A8	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_HT_MCS_2	A9	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_HT_MCS_3	AA	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_HT_MCS_4	AB	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_HT_MCS_5	AC	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
TX_PWR_HT_MCS_6	AD	RW	0	C6	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA
EXT_LNA_RX_GAIN	C0	RW	0	00	EXT_LNA		
EXT_LNA_RX_NF	C1	RW	0	00	EXT_LNA		
EXT_LNA_RX_P1DB	C2	RW	0	00	EXT_LNA		
EXT_LNA_BP_GAIN0	C3	RW	0	00	EXT_LNA		
EXT_LNA_BP_GAIN1	C4	RW	0	00	EXT_LNA		
EXT_LNA_BP_P1DB	C5	RW	0	00	EXT_LNA		
STEP_NUM_NEG_7	C6	RW	0	00	STEP_NUM		
STEP_NUM_NEG_6	C7	RW	0	00	STEP_NUM		
STEP_NUM_NEG_5	C8	RW	0	00	STEP_NUM		
STEP_NUM_NEG_4	C9	RW	0	1A	STEP_NUM		
STEP_NUM_NEG_3	CA	RW	0	22	STEP_NUM		
STEP_NUM_NEG_2	CB	RW	0	2A	STEP_NUM		
STEP_NUM_NEG_1	CC	RW	0	31	STEP_NUM		
STEP_NUM_NEG_0	CD	RW	0	35	STEP_NUM		
STEP_NUM_REF	CE	RW	0	01	STEP_REF		
STEP_NUM_TEMP	CF	RW	0	35	STEP_TEMP		
STEP_NUM_POS_1	D0	RW	0	39	STEP_NUM		
STEP_NUM_POS_2	D1	RW	0	40	STEP_NUM		
STEP_NUM_POS_3	D2	RW	0	46	STEP_NUM		
STEP_NUM_POS_4	D3	RW	0	4D	STEP_NUM		

STEP_NUM_POS_5	D4	**RW**	0	7F	STEP_NUM		
STEP_NUM_POS_6	D5	**RW**	0	7F	STEP_NUM		
STEP_NUM_POS_7	D6	**RW**	0	7F	STEP_NUM		
XTAL_CAL	F4	**RW**	0	C0	XTAL_CAP_VLD	XTAL_CAP	
XTAL_TRIM2	F5	**RW**	0	00	XTAL_TRIM2_EN	XTAL_TRIM2_DEC	XTAL_TRIM2
XTAL_TRIM3	F6	**RW**	0	00	XTAL_TRIM3_EN	XTAL_TRIM3_DEC	XTAL_TRIM3

2.1 Chip ID (0x00h)

0000 **CHIP_ID** **Chip Identifier** **7628**

Bit	15	14	13	12	11	10	9	8
Name	**CHIP_ID[15:8]**							
Type	DC							
Reset	0	1	1	1	0	1	1	0
Bit	7	6	5	4	3	2	1	0
Name	**CHIP_ID[7:0]**							
Type	DC							
Reset	0	0	1	0	1	0	0	0

Bit(s)	Name	Description
15:0	CHIP_ID	**Chip ID**

2.2 Layout Revision ID (0x02h)

0002 **EEPROM_REV** **EEPROM Revision** **0200**

Bit	15	14	13	12	11	10	9	8
Name	**REL_REV**							
Type	DC							
Reset	0	0	0	0	0	0	1	0
Bit	7	6	5	4	3	2	1	0
Name	**ENG_REV**							
Type	DC							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:8	REL_REV	**Revision for formal release**

Bit(s)	Name	Description
7:0	ENG_REV	**Revision for engineer sample**

2.3 WIFI MAC Address(0x04h)

0004	WLAN_MAC_B1B0	WLAN MAC Address Low Bye	0C00

Bit	15	14	13	12	11	10	9	8
Name	MAC_ADDR_L[15:8]							
Type	RW							
Reset	0	0	0	0	1	1	0	0
Bit	7	6	5	4	3	2	1	0
Name	MAC_ADDR_L[7:0]							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	MAC_ADDR_L	**WLAN MAC Address Byte 1/0**

0006	WLAN_MAC_B3B2	WLAN MAC Address Middle Byte	E143

Bit	15	14	13	12	11	10	9	8
Name	MAC_ADDR_M[15:8]							
Type	RW							
Reset	1	1	1	0	0	0	0	1
Bit	7	6	5	4	3	2	1	0
Name	MAC_ADDR_M[7:0]							
Type	RW							
Reset	0	1	0	0	0	0	1	1

Bit(s)	Name	Description
15:0	MAC_ADDR_M	**WLAN MAC Address Byte 3/2**

0008	WLAN_MAC_B5B4	WLAN MAC Address High Byte	2876

Bit	15	14	13	12	11	10	9	8
Name	MAC_ADDR_H[15:8]							
Type	RW							
Reset	0	0	1	0	1	0	0	0
Bit	7	6	5	4	3	2	1	0
Name	MAC_ADDR_H[7:0]							
Type	RW							

Reset	0	1	1	1	0	1	1	0

Bit(s)	Name	Description
15:0	MAC_ADDR_H	**WLAN MAC Address Byte 5/4**

2.4 WIFI MAC Address(0x28h)

0028 **MAC0_B1B0** **MAC 0 Address Low Bye** **0C00**

Bit	15	14	13	12	11	10	9	8
Name	MAC0_ADDR_L[15:8]							
Type	RW							
Reset	0	0	0	0	1	1	0	0
Bit	7	6	5	4	3	2	1	0
Name	MAC0_ADDR_L[7:0]							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	MAC0_ADDR_L	**MAC 0 (LAN) Address Byte 1/0**

002A **MAC0_B3B2** **MAC 0 Address Middle Byte** **E143**

Bit	15	14	13	12	11	10	9	8
Name	MAC0_ADDR_M[15:8]							
Type	RW							
Reset	1	1	1	0	0	0	0	1
Bit	7	6	5	4	3	2	1	0
Name	MAC0_ADDR_M[7:0]							
Type	RW							
Reset	0	1	0	0	0	0	1	1

Bit(s)	Name	Description
15:0	MAC0_ADDR_M	**MAC 0 (LAN) Address Byte 3/2**

002C **MAC0_B5B4** **MAC 0 Address High Byte** **2976**

Bit	15	14	13	12	11	10	9	8
Name	MAC0_ADDR_H[15:8]							
Type	RW							
Reset	0	0	1	0	1	0	0	1
Bit	7	6	5	4	3	2	1	0
Name	MAC0_ADDR_H[7:0]							
Type	RW							
Reset	0	1	1	1	0	1	1	0

Bit(s)	Name	Description
15:0	MAC0_ADDR_H	**MAC 0 (LAN) Address Byte 5/4**

2.5 WIFI MAC Address(0x2Eh)

002E	MAC1_B1B0	MAC 1 Address Low Bye	0C00

Bit	15	14	13	12	11	10	9	8
Name	MAC1_ADDR_L[15:8]							
Type	RW							
Reset	0	0	0	0	1	1	0	0
Bit	7	6	5	4	3	2	1	0
Name	MAC1_ADDR_L[7:0]							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	MAC1_ADDR_L	**MAC 1 (WAN) Address Byte 1/0**

0030	MAC1_B3B2	MAC 1 Address Middle Byte	E143

Bit	15	14	13	12	11	10	9	8
Name	MAC1_ADDR_M[15:8]							
Type	RW							
Reset	1	1	1	0	0	0	0	1
Bit	7	6	5	4	3	2	1	0
Name	MAC1_ADDR_M[7:0]							
Type	RW							
Reset	0	1	0	0	0	0	1	1

Bit(s)	Name	Description
15:0	MAC1_ADDR_M	**MAC 1 (WAN) Address Byte 3/2**

0032	MAC1_B5B4	MAC 1 Address High Byte	2A76

Bit	15	14	13	12	11	10	9	8
Name	MAC1_ADDR_H[15:8]							
Type	RW							
Reset	0	0	1	0	1	0	1	0
Bit	7	6	5	4	3	2	1	0
Name	MAC1_ADDR_H[7:0]							
Type	RW							
Reset	0	1	1	1	0	1	1	0

Bit(s)	Name	Description
15:0	MAC1_ADDR_H	**MAC 1 (WAN) Address Byte 5/4**

2.6 NIC Configuration 0 (0x34h)

0034	NIC_CONFG_0	NIC Configuration #0	3422

Bit	15	14	13	12	11	10	9	8
Name	EXT_PA_ANTSEL		BOARD_TYPE			EXT_PA_DRV	EXT_2P4G_PA	EXT_5G_PA
Type	RW		RW			RW	RW	RW
Reset	0	0	1	1		1	0	0
Bit	7	6	5	4	3	2	1	0
Name	TX_PATH				RX_PATH			
Type	RW				RW			
Reset	0	0	1	0	0	0	1	0

Bit(s)	Name	Description
15:14	EXT_PA_ANTSEL	**External PA ANTSEL Table**
13:12	BOARD_TYPE	**Board Type**
10	EXT_PA_DRV	**Exteral PA Driving** 0: 16mA 1: 8 mA
9	EXT_2P4G_PA	**External 2.4G PA Enable** 0: Disable 1: Enable
8	EXT_5G_PA	**External 5G PA Enable** 0: Disable 1: Enable
7:4	TX_PATH	**TX Path Setting** These fields are to provide the TX front-end architecture in the system. 0: Reserved. 1: 1 TX front-end in the system. 2: 2 TX front-end in the system. Other: Reserved
3:0	RX_PATH	**RX Path Setting** These fields are to provide the RX front-end architecture in the system. 0: Reserved. 1: 1 RX front-end in the system. 2: 2 RX front-end in the system. Other: reserved

2.7 NIC Configuration 1 (0x36h)

0036	NIC_CONFG_1	NIC	0000

Configuration #1

Bit	15	14	13	12	11	10	9	8
Name	DAC_TEST		TSSI_COMP	ANT_DIV_CTRL				BW_40M_2P4G
Type	RW		RW	RW				RW
Reset	0		0	0	0			0
Bit	7	6	5	4	3	2	1	0
Name	WPS				WF1_AUX	WF0_AUX	TX_POWER	HW_RADIO
Type	RW				RW	RW	RW	RW
Reset	0				0	0	0	0

Bit(s)	Name	Description
15	DAC_TEST	**DAC test bit**
13	TSSI_COMP	**TSSI power compensation enable** 0: disable TSSI power compensation , use per-channel ALC code 1: enable TSSI power compensation, TSSI slop offset scheme.
12:11	ANT_DIV_CTRL	**Antenna Diversity control** 00: disable diversity function (default value). 01: enable diversity function. 10: Fix antenna at main antenna 11: Fix antenna at auxiliary antenna
8	BW_40M_2P4G	**40M BW in 2.4GHz band** 0: enable 40MHz bandwidth for 2.4GHz band 1: disable 40MHz bandwidth for 2.4GHz band
7	WPS	**WPS Push Button Configuration control.** 0: disable WPS PBC control (default value). 1: enable WPS PBC control.
3	WF1_AUX	**WF1 Aux Rx path selection** 0: Use Main path Rx path, board select main rx path as rx data in. 1: Use Aux Rx path, need to set, board select aux rx path as rx data in. In this mode, FW also refer 0xC0~0xC5 as external LNA gain setting.
2	WF0_AUX	**WF0 Aux Rx path selection** 0: Use Main path Rx path, board select main rx path as rx data in. 1: Use Aux Rx path, need to set, board select aux rx path as rx data in. In this mode, FW also refer 0xC0~0xC5 as external LNA gain setting.
1	TX_POWER	**TX power temperature compensation scheme enable** This bit will disable/enable temperature compensation scheme. While this bit is enable, it means Tx power TSSI scheme is disable(0x37 bit5 = 0) and using per-channel Tx ALC code scheme. 0: disable temperature compensation 1: Enable temperature compensation
0	HW_RADIO	**HW Radio Control** When "hardware radio control" bit is enabled (=1), the driver will read MAC's GPIO[2] status. When GPIO[2] pin is low, the radio is disabled. When GPIO[2] pin is high, the radio is enabled. The Radio ON/OFF is controlled by both software UI and MAC's GPIO[2] pin. 0: disable hardware radio control (default value). 1: enable hardware radio control.

2.8 Country Region Code for 2.4G band (0x39h)

39	COUNTRY_REG	Country Region 2.4G Band	00

Bit	7	6	5	4	3	2	1	0
Name	BAND_2P4G							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	BAND_2P4G	**Country Region 2.4G Band** Default value = FFh, which means read from INF and registry, more flexible than reading from EEPROM, this is our current InstallShield CCS implementation. We do not recommend customers to read SKU from EEPROM. Value FFh is the default value. CountryCode - specify the domain code, can be FFh or one of the followings, 0: CH1 - 11 1: CH1 - 13 2: CH10 - 11 3: CH10 - 13 4: CH14 5: CH1 - 14 6: CH3 - 9 7: CH5 - 13 30: Manual Channel setting (Refer to 0x100~101h for detail) 31: CH1 - 14 (CH1 - 11 active scan, CH12 - 14 passive scan) 32: CH1 - 13 (CH1 - 11 active scan, CH12 ~ 13 passive scan) 33: 802.11b: CH1 to CH14 are active scan. 802.11g/n: CH1 to CH13 are active scan. CH14 is disallowed

Default value = FFh, which means read from INF and registry, more flexible than reading from EEPROM, this is our current InstallShield CCS implementation. We do not recommend customers to read SKU from EEPROM. Value FFh is the default value.

Notes: If set to read SKU from EEPROM, only available if 2.4G Country Region code registers are programmed.

2.9 LED Mode (0x3Bh)

3B	LED_MODE	LED Mode Setting	01

Bit	7	6	5	4	3	2	1	0
Name	LED_CTRL							
Type	RW							
Reset	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	LED_CTRL	**LED Control Modes**

2.10 NIC Configuration 2 (0x42h)

0042 **NIC_CONFG_2** **NIC Configuration #2** **0022**

Bit	15	14	13	12	11	10	9	8
Name					**TEMP_COMP**	**XTAL_OPT**		**ANT_DIV**
Type					RW	RW		RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
Name	**TX_STREAM**				**RX_STREAM**			
Type	RW				RW			
Reset	0	0	1	0	0	0	1	0

Bit(s)	Name	Description
11	TEMP_COMP	**25C Temper Compsation Disable** 1: disable 0: enable
10:9	XTAL_OPT	**XTAL Option**
8	ANT_DIV	**HW ANT Diversity** 0: Disable 1: Enable
7:4	TX_STREAM	**TX Stream** 1: 1 stream 2: 2 stream
3:0	RX_STREAM	**RX Stream** 1: 1 stream 2: 2 stream

2.11 RSSI Offset for 2.4G band (0x46h)

0046 **RSSI_OFST** **2.4G RSSI Offset** **0000**

Bit	15	14	13	12	11	10	9	8
Name	**RSSI1_OFST**							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Name	**RSSI0_OFST**							
Type	RW							

Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:8	RSSI1_OFST	**2.4 G RSSI 1 Offset**
7:0	RSSI0_OFST	**2.4 G RSSI 0 Offset**

2.12 20M/40M BW Power Delta for 2.4G band (0x50h)

50	TX_POWER_DELTA	20/40 BW TX Power Delta for 2.4G	82

Bit	7	6	5	4	3	2	1	0
Name	DELTA_EN	DELTA_INC	DELTA					
Type	RW	RW	RW					
Reset	1	0	0	0	0	0	1	0

Bit(s)	Name	Description
7	DELTA_EN	**Power Delta Enable Bit** 0: Disable 1: Enable
6	DELTA_INC	**Power Delta Increase** 0: Decrease 1: Increase
5:0	DELTA	**Delta Value** 000001: 0.5dBm 000010: 1dBm 000011: 1.5dBm 000100: 2dBm 000101: 2.5dBm 000110: 3dBm 000111: 3.5dBm 001000: 4dBm

Example:

The default calibrated TX power as followings with the TX power delta configuration is not enable.

- 40M BW TX power= 14dBm and 20M BW TX power = 14dBm

If want keep 20M BW TX power in 14dBm and reduce 40M BW TX power to 10dBm (delta=4dBm), set 50h = 88h (1000 1000).

2.13 Temp. Sensor Calibration (0x55h)

55	TEMP_SEN_CAL	Temperature	B0

Sensor Calibration

Bit	7	6	5	4	3	2	1	0
Name	TEMP_COMP_EN	THADC_SLOP						
Type	RW	RW						
Reset	1	0	1	1	0	0	0	0

Bit(s)	Name	Description
7	TEMP_COMP_EN	**Temperature Compensation Enable** [note] Calibration-free Field 0: Disable 1: Enable
6:0	THADC_SLOP	**THADC cal read out value** This is THADC read out value, this value should follow the temp. formulation to get the temperature. [note] Calibration-free Field

2.14 2.4G Tx0 Power Slope /offset (0x56h~0x57h)

56	TX0_PA_TSSI_LSB	TX0 PA TSSI slop and Offset	C0

Bit	7	6	5	4	3	2	1	0
Name	TX0_PA_TSSI_OFST				TX0_PA_TSSI_SLOP			
Type	RW				RW			
Reset	1	1	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	TX0_PA_TSSI_OFST	**TX0 2.4G TX PA TSSI_Offset[3:0]** [note] Calibration-free Field
3:0	TX0_PA_TSSI_SLOP	**TX0 2.4G TX PA TSSI Slop** [note] Calibration-free Field

57	TX0_PA_TSSI_MSB	TX0 PA TSSI Offset MSB	CC

Bit	7	6	5	4	3	2	1	0
Name	TX0_PA_TSSI_OFST							
Type	RW							
Reset	1	1	0	0	1	1	0	0

Bit(s)	Name	Description
7:0	TX0_PA_TSSI_OFST	**TX0 2.4G TX PA TSSI_Offset[11:4]**

Bit(s)	Name	Description
		[note] Calibration-free Field 0xCA: 7628 A/N 0xC8: 7628K

Driver compares current TSSI value with this TSSI reference value as a base to decide if real-time TX power compensation is required. 0xFF will be treated as invalid value. This function is controlled by 'external TX ALC' bit (NIC configuration1 bit1) or 'internal TX ALC ' bit (NIC configuration1 bit13).

2.15 2.4G Tx0 Target Power (0x58h)

58 **TX0_POWER** **TX0 TX Power** **23**

Bit	7	6	5	4	3	2	1	0
Name	TX0_TX_PWR							
Type	RW							
Reset	0	0	1	0	0	0	1	1

Bit(s)	Name	Description
7:0	TX0_TX_PWR	**TX0 2.4G TX power (54Mbps, dBm)**

2.16 2.4G Tx0 Power Low/Middle/High Channel (0x59h ~ 0x5Bh)

59 **TX0_PWR_OFST_L** **TX0 TX Power Offset Low** **00**

Bit	7	6	5	4	3	2	1	0
Name	TX0_TX_PWR_EN	TX0_TX_PWR_INC	TX0_TX_PWR_OFST_L					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	TX0_TX_PWR_EN	**TX0 2.4G TX power offset Enable** 0: Disable 1: Enable
6	TX0_TX_PWR_INC	**TX0 2.4G TX power offset Increase** 0: Decrease 1: Increase
5:0	TX0_TX_PWR_OFST_L	**TX0 2.4G TX power offset low(CH1~5)(delta,dB)**

5A	TX0_PWR_OFST_M	TX0 TX Power Offset Middle	00

Bit	7	6	5	4	3	2	1	0
Name	TX0_TX_PWR_EN	TX0_TX_PWR_INC	TX0_TX_PWR_OFST_M					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	TX0_TX_PWR_EN	**TX0 2.4G TX power offset Enable** 0: Disable 1: Enable
6	TX0_TX_PWR_INC	**TX0 2.4G TX power offset Increase** 0: Decrease 1: Increase
5:0	TX0_TX_PWR_OFST_M	**TX0 2.4G TX power offset middle(CH6~10)(delta,dB)**

5B	TX0_PWR_OFST_H	TX0 TX Power Offset High	00

Bit	7	6	5	4	3	2	1	0
Name	TX0_TX_PWR_EN	TX0_TX_PWR_INC	TX0_TX_PWR_OFST_H					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	TX0_TX_PWR_EN	**TX0 2.4G TX power offset Enable** 0: Disable 1: Enable
6	TX0_TX_PWR_INC	**TX0 2.4G TX power offset Increase** 0: Decrease 1: Increase
5:0	TX0_TX_PWR_OFST_H	**TX0 2.4G TX power offset high(CH11~14)(delta,dB)**

0x59~0x5B are used as channel TX power compensation in customer production line.
Customers could set different TX power compensation value according to different PCB design to reach flatter power responds.
For example
If customers found PCB had 1.5dB higher power variation in low channels and 1.5dB lower power variation in high channels.
Customer could use channel compensation offset to get flatter performance like setting as below.

Offset	Description	Example
0x59	TX0 2.4G Tx power offset low (CH1~5)(delta,dB)	0x83=> means SW will decrease 3 step(around -1.5dB) corresponding to TX0 2.4G TX power setting.
0x5A	TX0 2.4G Tx power offset middle (CH6~10)(delta,dB)	0x80=> means SW will decrease 0 step(around 0dB) corresponding to TX0 2.4G TX power setting.
0x5B	TX0 2.4G Tx power offset high (CH11~14)(delta,dB)	0xC3=> means SW will increase 3 step(around +1.5dB) corresponding to TX0 2.4G TX power setting.

2.17 2.4G Tx1 Power Slope /offset (0x5Ch~0x5Dh)

5C **TX1_PA_TSSI_LSB** **TX1 PA TSSI slop and Offset** **40**

Bit	7	6	5	4	3	2	1	0
Name	TX1_PA_TSSI_OFST				TX1_PA_TSSI_SLOP			
Type	RW				RW			
Reset	0	1	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	TX1_PA_TSSI_OFST	**TX1 2.4G TX PA TSSI Offset[3:0]** [note] Calibration-free Field
3:0	TX1_PA_TSSI_SLOP	**TX1 2.4G TX PA TSSI Slop** [note] Calibration-free Field

5D **TX1_PA_TSSI_MSB** **TX1 PA TSSI Offset MSB** **CC**

Bit	7	6	5	4	3	2	1	0
Name	TX1_PA_TSSI_OFST							
Type	RW							
Reset	1	1	0	0	1	1	0	0

Bit(s)	Name	Description
7:0	TX1_PA_TSSI_OFST	**TX1 2.4G TX PA TSSI Offset[11:4]** [note] Calibration-free Field

2.18 2.4G Tx1 Target Power (0x5Eh)

5E **TX1_POWER** **TX1 TX** **23**

Power

Bit	7	6	5	4	3	2	1	0
Name	TX1_TX_PWR							
Type	RW							
Reset	0	0	1	0	0	0	1	1

Bit(s)	Name	Description
7:0	TX1_TX_PWR	**TX1 2.4G TX power (54Mbps, dBm)**

2.19 2.4G Tx1 Power Offset Low/Middle/High Channel(0x5Fh~0x61h)

5F	TX1_PWR_OFST_L	TX1 TX Power Offset Low	00

Bit	7	6	5	4	3	2	1	0
Name	TX1_TX_PWR_EN	TX1_TX_PWR_INC	TX1_TX_PWR_OFST_L					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	TX1_TX_PWR_EN	**TX1 2.4G TX power offset Enable** 0: Disable 1: Enable
6	TX1_TX_PWR_INC	**TX1 2.4G TX power offset Increase** 0: Decrease 1: Increase
5:0	TX1_TX_PWR_OFST_L	**TX1 2.4G TX power offset low(CH1~5)(delta,dB)**

60	TX1_PWR_OFST_M	TX1 TX Power Offset Middle	00

Bit	7	6	5	4	3	2	1	0
Name	TX1_TX_PWR_EN	TX1_TX_PWR_INC	TX1_TX_PWR_OFST_M					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	TX1_TX_PWR_EN	**TX1 2.4G TX power offset Enable** 0: Disable 1: Enable
6	TX1_TX_PWR_INC	**TX1 2.4G TX power offset Increase** 0: Decrease 1: Increase

Bit(s)	Name	Description
5:0	TX1_TX_PWR_OFST_M	**TX1 2.4G TX power offset middle(CH6~10)(delta,dB)**

61	TX1_PWR_OFST_H	TX1 TX Power Offset High	00

Bit	7	6	5	4	3	2	1	0
Name	TX1_TX_PWR_EN	TX1_TX_PWR_INC	TX1_TX_PWR_OFST_H					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	TX1_TX_PWR_EN	**TX1 2.4G TX power offset Enable** 0: Disable 1: Enable
6	TX1_TX_PWR_INC	**TX1 2.4G TX power offset Increase** 0: Decrease 1: Increase
5:0	TX1_TX_PWR_OFST_H	**TX1 2.4G TX power offset high(CH11~14)(delta,dB)**

2.20 2.4G Tx rate power configuration (0xA0h~0xBFh)

A0	TX_PWR_CCK_0	2.4GHz TX Power for CCK 1M/2M	C6

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A1	TX_PWR_CCK_1	2.4GHz	C6

TX Power for CCK 5.5M/11M

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A2 **TX_PWR_OFDM_0** **2.4GHz TX Power for OFDM 6M/9M** **C6**

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A3 **TX_PWR_OFDM_1** **2.4GHz TX Power for OFDM 12M/18M** **C6**

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A4 **TX_PWR_OFDM_2** **2.4GHz TX Power for OFDM 24M/36M** **C6**

Bit	7	6	5	4	3	2	1	0
Name	**TX_PWR_COMP_EN**	**TX_PWR_INC**	**TX_PWR_DELTA**					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A5 **TX_PWR_OFDM_3** **2.4GHz TX Power for OFDM 48M** **C6**

Bit	7	6	5	4	3	2	1	0
Name	**TX_PWR_COMP_EN**	**TX_PWR_INC**	**TX_PWR_DELTA**					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A6 **TX_PWR_OFDM_4** **2.4GHz TX Power for OFDM 54M** **C6**

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A7 **TX_PWR_HT_MCS_0** **2.4GHz TX Power for HT MCS=0/8** **C6**

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A8 **TX_PWR_HT_MCS_1** **2.4GHz TX Power for HT MCS=32** **C6**

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

A9 **TX_PWR_HT_MCS_2** **2.4GHz TX Power for HT MCS=1,2/ 9,10** **C6**

Bit	7	6	5	4	3	2	1	0
Name	**TX_PWR_COMP_EN**	**TX_PWR_INC**	**TX_PWR_DELTA**					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

AA **TX_PWR_HT_MCS_3** **2.4GHz TX Power for HT MCS=3,4/ 11,12** **C6**

Bit	7	6	5	4	3	2	1	0
Name	**TX_PWR_COMP_EN**	**TX_PWR_INC**	**TX_PWR_DELTA**					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

AB	TX_PWR_HT_MCS_4	2.4GHz TX Power for HT MCS=5/13	C6

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

AC	TX_PWR_HT_MCS_5	2.4GHz TX Power for HT MCS=6/14	C6

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_COMP_EN	TX_PWR_INC	TX_PWR_DELTA					
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

AD	TX_PWR_HT_MCS_6	2.4GHz TX Power for HT MCS=7/15	C6

Bit	7	6	5	4	3	2	1	0
Name	TX_PWR_C	TX_PWR_I	TX_PWR_DELTA					

	OMP_EN	NC						
Type	RW	RW	RW					
Reset	1	1	0	0	0	1	1	0

Bit(s)	Name	Description
7	TX_PWR_COMP_EN	**Power Compesation Enable** 0: Disable 1: Enable
6	TX_PWR_INC	**Increase TX Power with the Delta Value** 0: Decrease 1: Increase
5:0	TX_PWR_DELTA	**TX Power Delta (delta, dB)**

Default value=0x00, 6bit signed 2's complement value. (1 step=0.5dBm)
0xA0~0xBE are used as TX rate power configuration in customer production line.
Customers could set different TX rate power according to different RF power requirement.

Example:

If the table content is :

Offset	Ex. Value	Description	Example description
A0h	C3	2G TX power for CCK 1M/2M	0xC3=> 2G 1~11M & 6~18M will have 1.5dB higher power than 54M. 0x00=> 2G 24~54M will have equal power with 54M.
A1h	C3	2G TX power for CCK 5.5M/11M	
A2h	C3	2G TX power for OFDM 6M/9M	
A3h	C3	2G TX power for OFDM 12M/18M	
A4h	0	2G TX power for OFDM 24M/36M	
A5h	0	2G TX power for OFDM 48M	
A6h	0	2G TX power for OFDM 54M	
A7h	C2	2G TX power for HT/VHT MCS=0/8	0xC2 => 2G HT MCS0~3 &MCS8~11 will have 1dB higher power than 54M. 0x82 => 2G HT MCS4~7 &MCS12~15 will have 1dB lower power than 54M.
A8h	C2	2G TX power for HT/VHT MCS=32	
A9h	82	2G TX power for HT/VHT MCS=1,2/9,10	
AAh	82	2G TX power for HT/VHT MCS=3,4/11,12	0xC2=> 5G HT MCS0~3 &MCS8~11 will have 1dB higher power than 54M. 5G VHT MCS0~3 will have 1dB higher power than 54M. 0x82=> 5G HT MCS4~7 &MCS12~15 will have 1dB lower power than 54M. 5G VHT MCS4~7 will have 1dB lower power than 54M.
ABh	C2	2G TX power for HT MCS=5/13	
ACh	C2	2G TX power for HT MCS=6/14	
ADh	82	2G TX power for HT MCS=7/15	

2.21 External LNA (0xC0h)

C0	EXT_LNA_RX_GAIN	External LNA RX	00

GAIN

Bit	7	6	5	4	3	2	1	0
Name	EXT_LNA							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	EXT_LNA	**External LNA**

C1	EXT_LNA_RX_NF	External LNA RX NF	00

Bit	7	6	5	4	3	2	1	0
Name	EXT_LNA							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	EXT_LNA	**External LNA**

C2	EXT_LNA_RX_P1DB	External LNA RX P1 DB	00

Bit	7	6	5	4	3	2	1	0
Name	EXT_LNA							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	EXT_LNA	**External LNA**

C3	EXT_LNA_BP_GAIN0	External LNA BYPASS RX GAIN 0	00

Bit	7	6	5	4	3	2	1	0
Name	EXT_LNA							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	EXT_LNA	**External LNA**

C4	EXT_LNA_BP_GAIN1	External LNA BYPASS RX GAIN 1	00

Bit	7	6	5	4	3	2	1	0
Name	EXT_LNA							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	EXT_LNA	External LNA

CODA_REG_EXT_LNA_BP_P2DB

2.22 2.4GHz Step Number (0xC6h)

C6	STEP_NUM_NEG_7	Step Number for -7	00

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	STEP_NUM	Step Number for -7

C7	STEP_NUM_NEG_6	Step Number for -6	00

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	STEP_NUM	Step Number for -6

C8	STEP_NUM_NEG_5	Step Number for -5	00

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							

Type	RW							
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for -5**

C9 **STEP_NUM_NEG_4** **Step Number for -4** **1A**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_NUM**							
Type	RW							
Reset	0	0	0	1	1	0	1	0

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for -4**

CA **STEP_NUM_NEG_3** **Step Number for -3** **22**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_NUM**							
Type	RW							
Reset	0	0	1	0	0	0	1	0

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for -3**

CB **STEP_NUM_NEG_2** **Step Number for -2** **2A**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_NUM**							
Type	RW							
Reset	0	0	1	0	1	0	1	0

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for -2**

CC **STEP_NUM_NEG_1** **Step Number for -1** **31**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_NUM**							

Type	RW							
Reset	0	0	1	1	0	0	0	1

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for -1**

CD **STEP_NUM_NEG_0** **Step Number for -0** **35**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_NUM**							
Type	RW							
Reset	0	0	1	1	0	1	0	1

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for -0**

CE **STEP_NUM_REF** **2.4GHz Reference Step** **01**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_REF**							
Type	RW							
Reset	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	STEP_REF	**2.4GHz Reference Step**

CF **STEP_NUM_TEMP** **2.4GHz Reference Temperature** **35**

Bit	7	6	5	4	3	2	1	0
Name	**STEP_TEMP**							
Type	RW							
Reset	0	0	1	1	0	1	0	1

Bit(s)	Name	Description
7:0	STEP_TEMP	**2.4GHz Reference Temperature**

D0 **STEP_NUM_POS_1** **Step Number for +1** **39**

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	0	1	1	1	0	0	1

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for +1**

D1 **STEP_NUM_POS_2** **Step Number for +2** **40**

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	1	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for +2**

D2 **STEP_NUM_POS_3** **Step Number for +3** **46**

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	1	0	0	0	1	1	0

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for +3**

D3 **STEP_NUM_POS_4** **Step Number for +4** **4D**

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	1	0	0	1	1	0	1

Bit(s)	Name	Description
7:0	STEP_NUM	**Step Number for +4**

D4 **STEP_NUM_POS_5** **Step Number for +5** **7F**

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	STEP_NUM	Step Number for +5

D5	STEP_NUM_POS_6	Step Number for +6	7F

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	STEP_NUM	Step Number for +6

D6	STEP_NUM_POS_7	Step Number for +7	7F

Bit	7	6	5	4	3	2	1	0
Name	STEP_NUM							
Type	RW							
Reset	0	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	STEP_NUM	Step Number for +7

2.23 Frequency offset (0xF4h ~ 0xF6h)

F4	XTAL_CAL	Frequency Offset (XTAL Calibration)	Co

Bit	7	6	5	4	3	2	1	0
Name	XTAL_CAP_VLD	XTAL_CAP						
Type	RW	RW						
Reset	1	1	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
7	XTAL_CAP_VLD	**XTAL Cpa. Code Valid** [note] Calibration-free Field
6:0	XTAL_CAP	**XTAL Cap. Code** [note] Calibration-free Field

F5	XTAL_TRIM2	XTAL Trim 2 Compenaton (on RFB)	00

Bit	7	6	5	4	3	2	1	0
Name	XTAL_TRIM2_EN	XTAL_TRIM2_DEC	XTAL_TRIM2					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	XTAL_TRIM2_EN	**XTAL Trim 2 Enable** 0: Disable 1: Enable
6	XTAL_TRIM2_DEC	**XTAL Trim 2 Decrease** 0: Increase 1: Decrease
5:0	XTAL_TRIM2	**XTAL Trim 2 Value**

F6	XTAL_TRIM3	XTAL Trim 3 Compenaton	00

Bit	7	6	5	4	3	2	1	0
Name	XTAL_TRIM3_EN	XTAL_TRIM3_DEC	XTAL_TRIM3					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	XTAL_TRIM3_EN	**XTAL Trim 3 Enable** 0: Disable 1: Enable
6	XTAL_TRIM3_DEC	**XTAL Trim 3 Decrease** 0: Increase 1: Decrease
5:0	XTAL_TRIM3	**XTAL Trim 3 Value**

0xF4 is used for MTK FT test only for crystal calibration-free feature .

MTK wafer manufactory used 0xF4, bit 0~6, to store frequency offset value which is measured under MTK FT environment. Each IC has each corresponding frequency offset .

Bit 7 of 0xF4 is used to enable to apply crystal code vale. “1” means enable and “0” means disablle. While Bit7 is 0 (disable), it means rom code will not use crystal value of 0xF4 but use crystal code default value in rom code. Default is “1” (enable).

0xF5/0xF6 is used for crystal re-calibration purpose in customer production line

If customers want to re-do frequency trimming in customer production line, please use 0xF5/F6 as second /third frequency offset. Rom/Firmware code will check 0xF5/0xF6 Bit7 to decide the crystal trim code need to be compensated or not. Here is the formula :

```
If (0xF4[7] == 1 && 0xF5[7] == 1 && 0xF6[7] == 1)
   Final xtal trim code = 0xF4[6:0] +/- 0xF5[5:0] +/- 0xF6[5:0];
   // the increase/descrease(+/-) depends on 0xF5/F6[6]'s value
Else if(   (0xF4[7] == 1 && 0xF5[7] == 1 )
   Final xtal trim code = 0xF4[6:0] +/- 0xF5[5:0];
Else if(  (0xF4[7] == 1)
   Final xtal trim code = 0xF4[6:0];
Else
   Use rom code default vale.
```

2.24 Reserved for Customer (0x140h~0x1EFh)

MT7628

PROGRAMMING GUIDE

Version: 1.0
Release date: 2014-06-03

MT7628 Overview

The MT7628 SoC includes a high performance 580/575 MHz MIPS24KEc CPU core and high speed USB2.0/PCIe interfaces, which is designed to enable a multitude of high performance, cost-effective IEEE 802.11n applications with a MediaTek WiFi client card.

Functional Block Diagram

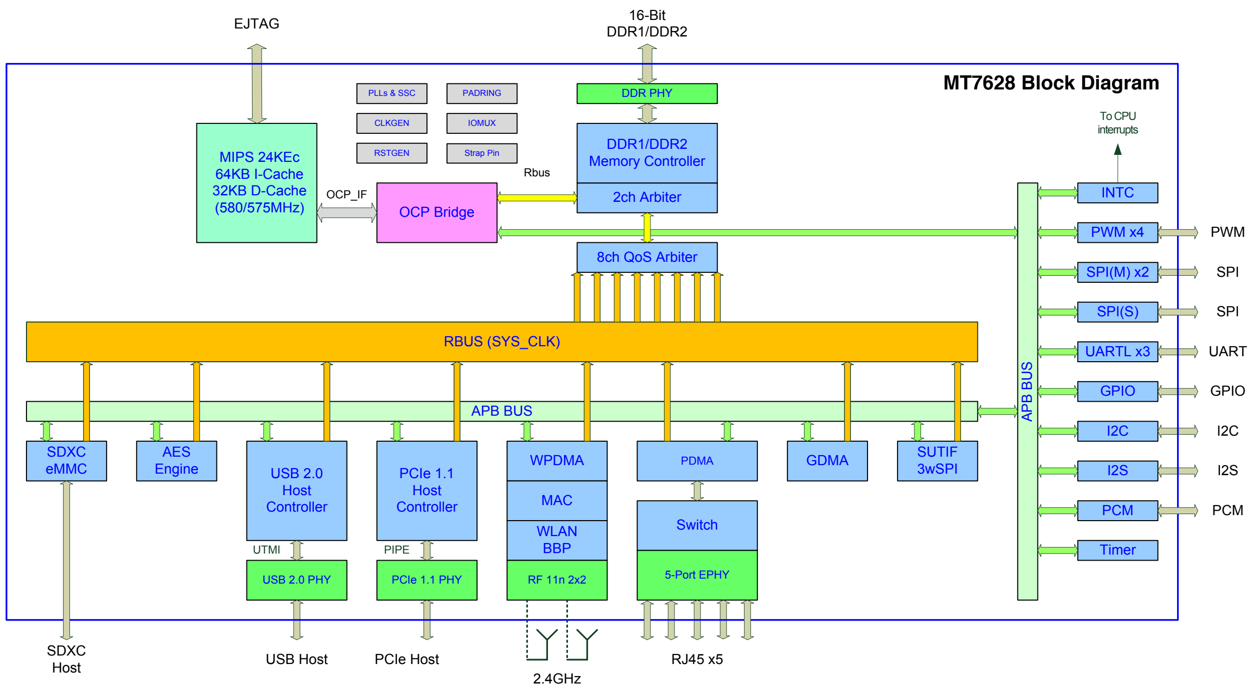


Figure 1-1 MT7628 Block Diagram

There are several masters (MIPS 24KEc, USB, PCI Express, SDXC, FE) in the MT7628 SoC on a high performance, low latency Rbus. In addition, the MT7628 SoC supports lower speed peripherals such as UART Lite, GPIO, I2C and SPI via a low speed peripheral bus (Pbus). The DDR/DDR2 controller is the only bus slave on the Rbus. It includes an Advanced Memory Scheduler to arbitrate the requests from bus masters, enhancing the performance of memory access intensive tasks.

Document Revision History

Revision	Date	Author	Description
1.0	2014-04-28	PeterCT Wu	Initial Draft

Table of Contents

Table of Figures

List of Tables

1. MIPS 24KEc Processor

1.1 Features

- 8-stage pipeline
- 32-bit address paths
- 64-bit data paths to caches and external interfaces
- MIPS32-Compatible Instruction Set
  - Multiply-Accumulate and Multiply-Subtract Instructions (MADD, MADDU, MSUB, MSUBU)
  - Targeted Multiply Instruction (MUL)
  - Zero/One Detect Instructions (CLZ, CLO)
  - Wait instructions (WAIT)
  - Conditional Move instructions (MOVZ, MOVN)
  - Prefetch instructions (PREF)
- MIPS32 Enhanced Architecture (Release 2) Features
  - Vectored interrupts and support for an external interrupt controller
  - Programmable exception vector base
  - Atomic interrupt enable/disable
  - GPR shadow registers (one, three or seven additional shadows can be optionally added to minimize latency for interrupt handlers)
  - Bit field manipulation instructions
- MIPS32 Privileged Resource Architecture
  - MIPS DSP ASE
  - Fractional data types (Q15, Q31)
  - Saturating arithmetic
  - SIMD instructions operate on 2x16 b or 4x8 b simultaneously
  - 3 additional pairs of accumulator registers
- Programmable Memory Management Unit
  - 32 dual-entry JTLB with variable page sizes
  - 4-entry ITLB
  - 8-entry DTLB
  - Optional simple Fixed Mapping Translation (FMT) mechanism
- MIPS16e™ Code Compression
  - 16-bit encodings of 32-bit instructions to improve code density
  - Special PC-relative instructions for efficient loading of addresses and constants
  - SAVE & RESTORE macro instructions for setting up and tearing down stack frames within subroutines
  - Improved support for handling 8 and 16-bit datatypes
- Programmable L1 Cache Sizes
  - Instruction cache size: 32 KB
  - Data cache size: 16 KB
- 4-Way Set Associative
  - Up to 8 outstanding load misses
  - Write-back and write-through support
  - 32-byte cache line size

1.2 Block Diagram

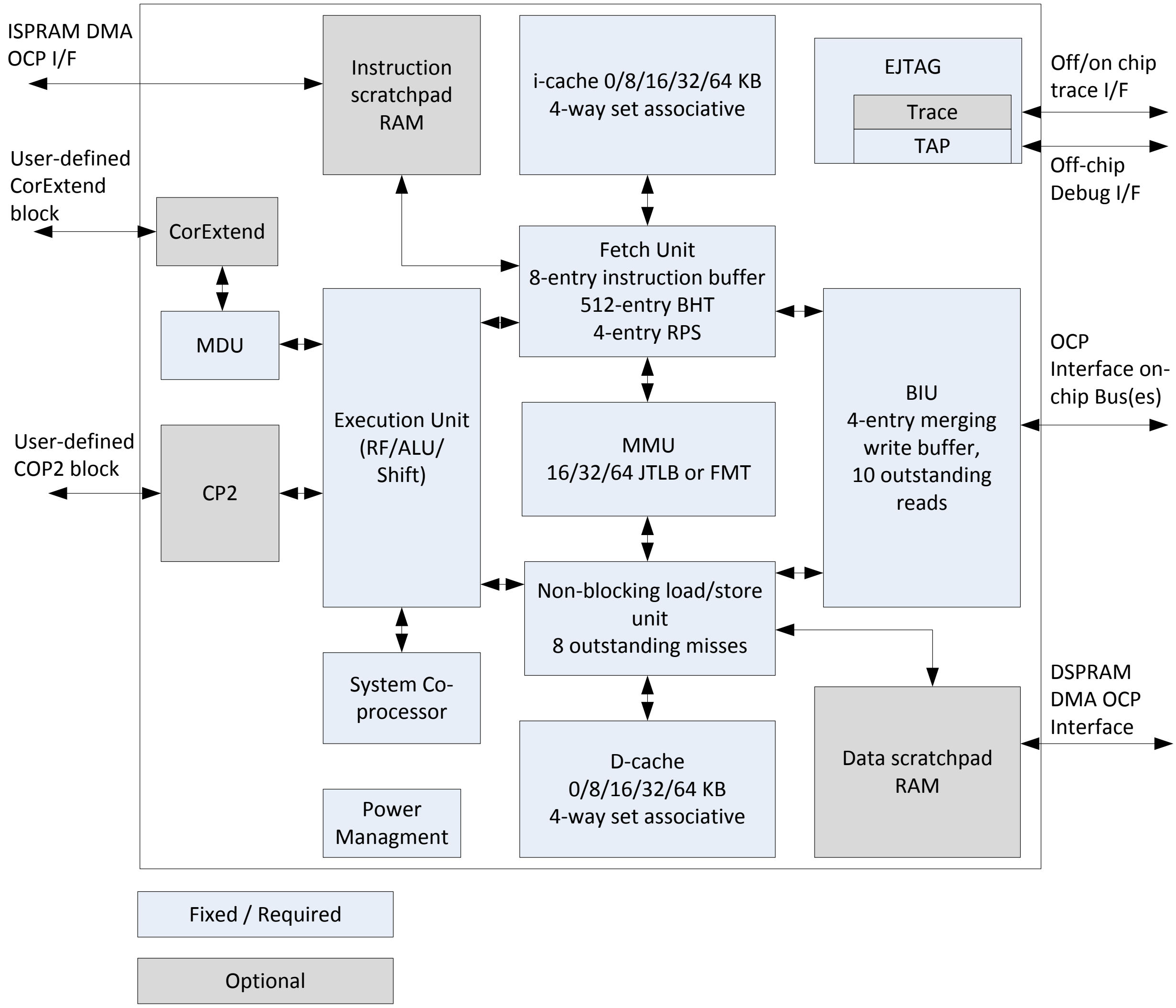


Figure 1-1 MIPS 24KEc CPU Block Diagram

1.3 Memory Map Summary

Start		End	Size	Description
0000.0000	-	0FFF.FFFF	256 MBytes	DDR 256 MB
1000.0000	-	1000.00FF	256 Bytes	SYSCTL
1000.0100	-	1000.01FF	256 Bytes	TIMER
1000.0200	-	1000.02FF	256 Bytes	INTCTL
1000.0300	-	1000.03FF	256 Bytes	EXT_MC_ARB (DDR/DDR II)
1000.0400	-	1000.04FF	256 Bytes	Rbus Matrix CTRL
1000.0500	-	1000.05FF	256 Bytes	MIPS CNT
1000.0600	-	1000.06FF	256 Bytes	GPIO
1000.0700	-	1000.07FF	256 Bytes	SPI Slave
1000.0800	-	1000.08FF	256 Bytes	<<Reserved>>
1000.0900	-	1000.09FF	256 Bytes	I2C
1000.0A00	-	1000.0AFF	256 Bytes	I2S
1000.0B00	-	1000.0BFF	256 Bytes	SPI Master
1000.0C00	-	1000.0CFF	256 Bytes	UARTLITE 1
1000.0D00	-	1000.0DFF	256 Bytes	UARTLITE 2
1000.0E00	-	1000.0EFF	256 Bytes	UARTLITE 3
1000.0F00	-	1000.0FFF	256 Bytes	<<Reserved>>
1000.1000	-	1000.17FF	2 KBytes	RGCTL
1000.1800	-	1000.1FFF	2 KBytes	<<Reserved>>
1000.2000	-	1000.27FF	2 KBytes	PCM (up to 16 channels)
1000.2800	-	1000.2FFF	2 KBytes	Generic DMA (up to 16 channels)
1000.3000	-	1000.3FFF	4 KBytes	<<Reserved>>
1000.4000	-	1000.4FFF	4 KBytes	AES Engine
1000.5000	-	1000.5FFF	4 Kbytes	PWM
1000.6000	-	100F.FFFF		<<Reserved>>
1010.0000	-	1010.FFFF	64 Kbytes	Frame Engine
1011.0000	-	1011.7FFF	32 KBytes	Ethernet Switch
1011.8000	-	1011.FFFF	32 KBytes	<<Reserved>>
1012.0000		1012.7FFF	32 KBytes	USB PHY
1012.8000	-	1012.FFFF	32 KBytes	<<Reserved>>
1013.0000	-	1013.7FFF	32 KBytes	SDXC / eMMC
1013.8000	-	1013.FFFF	32 KBytes	<<Reserved>>
1014.0000	-	1017.FFFF	256 KBytes	PCI Experss
1018.0000	-	101B.FFFF	256 KBytes	<<Reserved>>
101C.0000	-	101F.FFFF	256 KBytes	USB Host Controller
1020.0000	-	102F.FFFF	1 MBytes	<<Reserved>>
1030.0000	-	103F.FFFF	1 MBytes	WLAN MAC/BBP
1040.0000	-	1BFF.FFFF		<<Reserved>>
1C00.0000	-	1C3F.FFFF	4 MBytes	SPI Flash Direct Access
1C40.0000	-	1FFF.FFFF		<<Reserved>>
2000.0000	-	2FFF.FFFF	256 MBytes	PCIE Direct Access
3000.9999	-	3FFF.FFFF		<<Reserved>>

1.3 Interupt Table Summary

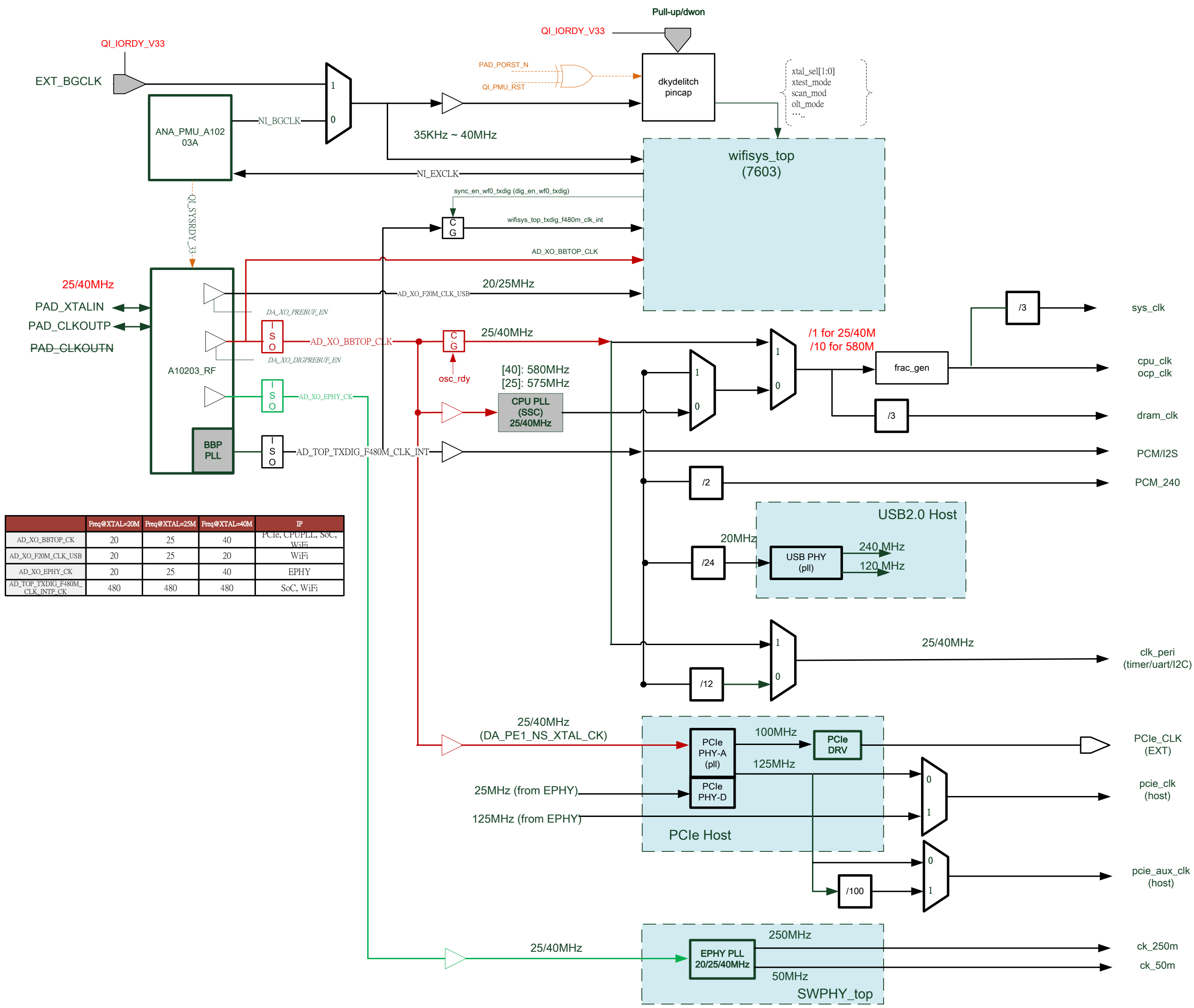
SI_Int -

	Module	Source Pin	Level/Edge
SI_Int0	soc_cirq	cpu_irq0	Level
SI_Int1	soc_cirq	cpu_irq1	Level
SI_Int2	PCIE	pcie_int_req	Level
SI_Int3	FE	fe_int_req	Level
SI_Int4	WLAN	wlan_int_req	Level
SI_Int5	MIPS24Kec/aux_tick	SI_TIMERInt/stk_int	Level

INTC -

	Module	Source Pin	Level/Edge
soc_cirq_int0	SYSCTL	sysctl_int	Level
soc_cirq_int1	SPIS	SW interrupt	Level
soc_cirq_int2			
soc_cirq_int3	DRAMC	mc_int	Level
soc_cirq_int4	PCM	pcm_int	Level
soc_cirq_int5			
soc_cirq_int6	GPIO	gpio_int	Level
soc_cirq_int7	GDMA	gdma_int	Level
soc_cirq_int8			
soc_cirq_int9	MIPS24Kec	pc_int	Level
soc_cirq_int10	I2S	i2s_int	Level
soc_cirq_int11	SPI	spi_int	Level
soc_cirq_int12			
soc_cirq_int13	AES	aes_int	Level
soc_cirq_int14	SDXC	sdxc_int	Level
soc_cirq_int15	PCTRL	r2p_int	Level
soc_cirq_int16	PCIE	pcie_link_down_rst_int	Level
soc_cirq_int17	ESW	esw_int	Level
soc_cirq_int18	USB20	uhstl_int	Level
soc_cirq_int19			
soc_cirq_int20	UART-LITE	uart0_int	Level
soc_cirq_int21	UART-LITE	uart1_int	Level
soc_cirq_int22	UART-LITE	uart2_int	Level
soc_cirq_int23	TIMER	wdtimer_int	Level
soc_cirq_int24	TIMER	timer0_int	Level
soc_cirq_int25	TIMER	timer1_int	Level
soc_cirq_int26	PWM	pwm_irq	Level
soc_cirq_int27	WLAN	wlan_wakeup_int	Level
soc_cirq_int28			
soc_cirq_int29			
soc_cirq_int30			
soc_cirq_int31			

1.4 Clock Plan



	Freq@XTAL=20M	Freq@XTAL=25M	Freq@XTAL=40M	IP
AD_XO_BBTOP_CK	20	25	40	PCIe, CPUPLL, SoC, WiFi
AD_XO_F20M_CLK_USB	20	25	20	WiFi
AD_XO_EPHY_CK	20	25	40	EPHY
AD_TOP_TXDIG_F480M_CLK_INTP_CK	480	480	480	SoC, WiFi

Figure 1-2 MT7628 Clock Diagram

2. Registers

2.1 Nomenclature

The following nomenclature is used for register types:

RO	Read Only
WO	Write Only
RW	Read or Write
RC	Read Clear
W1C	Write One Clear
-	Reserved bit
X	Undefined binary value

2.2 System Control

2.2.1 Features

– Provides read-only chip revision registers
– Provides a window to access boot-strapping signals
– Supports memory remapping configurations
– Supports software reset to each platform building block
– Provides registers to determine GPIO and other peripheral pin muxing schemes
– Provides some power-on-reset only test registers for software programmers
– Combines miscellaneous registers (such as clock skew control, status register, memo registers, etc)

2.2.2 Block Diagram

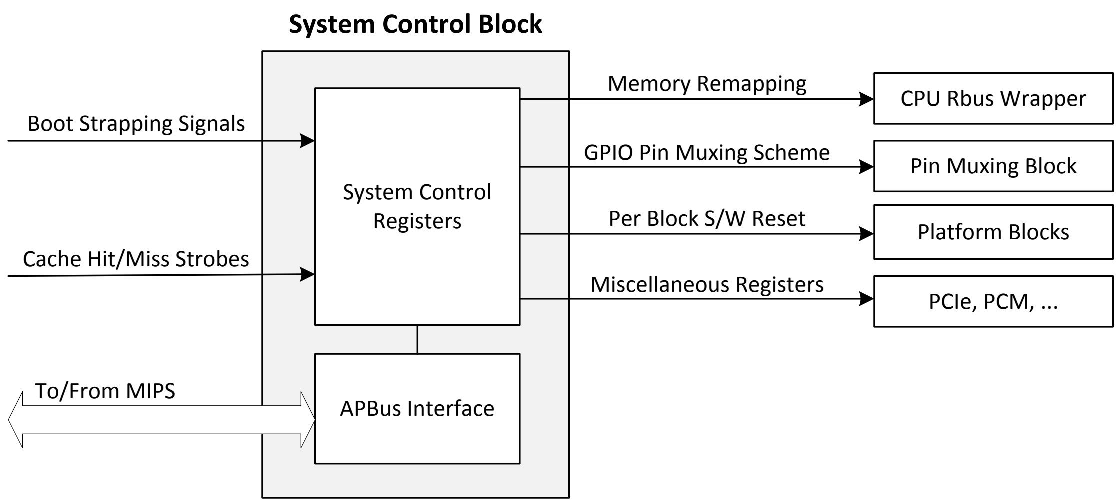


Figure 2-1 System Control Block Diagram

2.2.3 Registers

SYSCTL Changes LOG

Revision	Date	Author	Change Log
0.1	2013/10/3	PeterCT Wu	Initial for MT7628
0.2	2014/4/28	PeterCT Wu	MT7628 E2

Module name: SYSCTL Base address: (+10000000h)

Address	Name	Width	Register Function
10000000	CHIPID0_3	32	CHIP ID ASCII Character 0-3
10000004	CHIPID4_7	32	CHIP ID ASCII Character 4-7
10000008	EE_CFG	32	E-Fuse Configuration
1000000C	CHIP_REV_ID	32	Chip Revision Identification
10000010	SYSCFG0	32	System Configuration Register 0
10000014	SYSCFG1	32	System Configuration Register 1
10000018	TESTSTAT	32	Firmware Test Status
1000001C	TESTSTAT2	32	Firmware Test Status 2
10000028	ROM_STATUS	32	Andes ROM Status
1000002C	CLKCFG0	32	Clock Configuration Register 0
10000030	CLKCFG1	32	Clock Configuration Register 1
10000034	RSTCTL	32	Reset Control Register
10000038	RSTSTAT	32	Reset Status Register
1000003C	AGPIO_CFG	32	Analog GPIO Configuration
10000040	N9_GPIO_INT	32	Andes GPIO Interrupt
10000044	N9_GPIO_MASK	32	Andes GPIO Mask
10000060	GPIO1_MODE	32	GPIO1 purpose selection
10000064	GPIO2_MODE	32	GPIO2 purpose selection
10000068	MEMO1	32	Memory1
1000006C	MEMO2	32	Memory2
10000070	EXT_MEMO1	32	Extend Application #1
10000074	EXT_MEMO2	32	Extend Application #2
10000078	EXT_MEMO3	32	Extend Application #3
1000007C	EXT_MEMO4	32	Extend Application #4

10000000 CHIPID0_3 CHIP ID ASCII Character 0-3 3637544D

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CHIP_ID3								CHIP_ID2							
Type	RO								RO							
Reset	0	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHIP_ID1								CHIP_ID0							
Type	RO								RO							
Reset	0	1	0	1	0	1	0	0	0	1	0	0	1	1	0	1

Bit(s)	Name	Description

Bit(s)	Name	Description
31:24	CHIP_ID3	**ASCII CHIP Name Identification Character 3**
23:16	CHIP_ID2	**ASCII CHIP Name Identification Character 2**
15:8	CHIP_ID1	**ASCII CHIP Name Identification Character 1**
7:0	CHIP_ID0	**ASCII CHIP Name Identification Character 0**

10000004 **CHIPID4_7** **CHIP ID ASCII Character 4-7** **20203832**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CHIP_ID7								CHIP_ID6							
Type	RO								RO							
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CHIP_ID5								CHIP_ID4							
Type	RO								RO							
Reset	0	0	1	1	1	0	0	0	0	0	1	1	0	0	1	0

Bit(s)	Name	Description
31:24	CHIP_ID7	**ASCII CHIP Name Identification Character 3**
23:16	CHIP_ID6	**ASCII CHIP Name Identification Character 2**
15:8	CHIP_ID5	**ASCII CHIP Name Identification Character 1**
7:0	CHIP_ID4	**ASCII CHIP Name Identification Character 0**

10000008 **EE_CFG** **E-Fuse Configuration** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EE_CFG1															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EE_CFG0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	EE_CFG1	**E-Fuse Configuration 1**
15:0	EE_CFG0	**E-Fuse Configuration 0**

1000000C **CHIP_REV_ID** **Chip Revision Identification** **00010102**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																PKG_ID
Type																RO
Reset																1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					VER_ID								ECO_ID			
Type					RO								RO			
Reset					0	0	0	1					0	0	1	0

Bit(s)	Name	Description
16	PKG_ID	**Package ID** 0: DRQFN10x10-110 1: DRQFN12x12-156
11:8	VER_ID	**Chip Version ID**
3:0	ECO_ID	**Chip ECO ID**

10000010 SYSCFG0 System Configuration Register 0 00000100

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TEST_CODE**											**BS_SHADOW[8:4]**				
Type	RW											RO				
Reset	0	0	0	0	0	0	0	0				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BS_SHADOW[3:0]**							**DBG_JTAG_MODE**	**TEST_MODE_1**	**XTAL_FREQ_SEL**	**EXT_BG**	**TEST_MODE_0**	**CHIP_MODE**			**DRAM_TYPE**
Type	RO							RO	RO	RO	RO	RO	RO			RO
Reset	0	0	0	0				1	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	TEST_CODE	**Default value is from bootstrap and can be modified by software.**
20:12	BS_SHADOW	**BS shadow register for last boot-up value (by manual boot-strap SYSCFG1.PULL_EN)** Displays a backup copy of the last bootup value
8	DBG_JTAG_MODE	**JTAG for MIPS and Andes** 1: Normal Boot-up 0: JTAG mode(MIPS & Andes)
7	TEST_MODE_1	**Test Mode[1:0]**
6	XTAL_FREQ_SEL	**XTAL Frequency Selection** 0: 25MHz DIP 1: 40MHz SMD (3225)
5	EXT_BG	**External BG Clock** 0: BG clock from PMU 1: BG clock from the external pin
4	TEST_MODE_0	**Test Mode[1:0]** 0: SUTIF 1: 3-wire SPI
3:1	CHIP_MODE	Chip Mode A vector to set chip function/test/debug modes in non-test/debug operation. For more information see the Bootstrapping Pins Description in the datasheet for this chip. 000: Boot from PLL (boot from SPI 3-Byte ADR) 001: Boot from PLL (boot from SPI 4-Byte ADR) 010: Boot from XTAL (boot from SPI 3-Byte ADR) 011: Boot from XTAL (boot from SPI 4-Byte ADR) 100: SCAN mode 101: IDDQ mode 110: Power-On mode 111: UTIF test mode

Bit(s)	Name	Description
0	DRAM_TYPE	**DDR type** [note] This DDR attribute is not valid for KN package.. (7628KN has DDR1 KGD) 0: DDR2 1: DDR1

10000014 SYSCFG1 System Configuration Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																PULL_EN
Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

Bit(s)	Name	Description
16	PULL_EN	**Internal Manual Boot-Strap** 1: enable 0: disable

10000018 TESTSTAT Firmware Test Status 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TESTSTAT[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TESTSTAT[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TESTSTAT	**Firmware Test Status register** NOTE: This register is reset only by a power-on reset.

1000001C TESTSTAT2 Firmware Test Status 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TESTSTAT2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TESTSTAT2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:0	TESTSTAT2	**Firmware Test Status Register 2** NOTE: This register is reset only by a power-on reset.

10000028 **ROM_STATUS** **Andes ROM Status** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									STATUS							
Type									RO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	STATUS	**Andes ROM Status** 0: Power-on default 1: ROM initialization done 2: Wifi driver loaded

1000002C **CLKCFG0** **Clock Configuration Register 0** **00201000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			OSC_1US_DIV							INT_CLK_FDIV						INT_CLK_FFRAC[4:4]
Type			RW							RW						RW
Reset			0	0	0	0	0	0		0	1	0	0	0		0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INT_CLK_FFRAC[3:0]				REFCLK0_RATE				DIS_N9		PCIE_EXT_125M	PERI_CLK_SEL	DIS_BBP_SLEEP	EN_BBP_CLK	CPU_FRM_BBP	CPU_FRM_XTAL
Type	RW				RW				RW		RW	RW	RW	RW	RW	RW
Reset	0	0	0	1	0	0	0		0		0	0	0	0	0	0

Bit(s)	Name	Description
29:24	OSC_1US_DIV	**Oscillator 1 usec Divider** Sets the maximum for the reference clock counter for either a 20 MHz or 40 MHz external XTAL input. The count increments each 1usec (indicating 1 MHz), up to the maximum, before resetting to zero. This counts the frequency of an external XTAL. This count is used to output a 32 KHz frequency to the REFCLK0 pin. 0: Automatically generates a 1 usec system tick regardless of whether XTAL frequency is 20 MHz or 40 MHz. 39: Default value for an external 40 MHz XTAL. 19: Default value for an external 20 MHz XTAL. Others: Manual mode for tick generation.
22:18	INT_CLK_FDIV	**Internal Clock Frequency Divider for I2S/PCM** The frequency divider used to generate the Fraction-N clock frequency.

Bit(s)	Name	Description
		Valid values range from 1 to 31. Fraction-N clock frequency = (INT_CLK_FFRAC/INT_CLK_FDIV)*PLL_FREQ
16:12	INT_CLK_FFRAC	**Internal Clock Fraction-N Frequency for I2S/PCM** A parameter used in conjunction with INT_CLK_FDIV to generate the Fraction-N clock frequency. Valid values range from 0 to 31. Fraction-N clock Frequency = (INT_CLK_FFRAC/INT_CLK_FDIV)*PLL_FREQ
11:9	REFCLK0_RATE	**Output clock rate of reference Clock 0** 7: CPUPLL Clock/8 6: Off 5: Internal Fraction-N_CLK/2 (I2S/PCM) 4: 48 MHz 3: 40 MHz 2: 25 MHz 1: 12 MHz 0: Xtal clock(25/40 MHz by boot strap)
7	DIS_N9	**Pause Andes Execution** [Note] This bit is initialized by HW STRAP and can be changed by SW afterwards. 1: Enable 0: default
5	PCIE_EXT_125M	**PCIe 125MHZ Clock Source** 1: Ext. 125MHz Source (EPHY) 0: PCIe PHY 125M
4	PERI_CLK_SEL	**Peripheral Clock Source Select** 1: XTAL input 0: 40 MHz from BBP 480 MHz divided by 12
3	DIS_BBP_SLEEP	**BBPPLL Sleep Mode Control** 1: Disable BBPPLL entering SLEEP mode 0: BBPPLL SLEEP mode
2	EN_BBP_CLK	**BBPPLL 480MHz Clock** 1: BBPPLL Clock Enable 0: BBPPLL Clock Disable
1	CPU_FRM_BBP	**CPU clock from BBPPLL** 1: 480MHz BBPPLL 0: 580MHz CPUPLL
0	CPU_FRM_XTAL	**CPU clock from XTAL** [Note] This bit is initialized by HW STRAP and can be changed by SW afterwards. 1: XTAL input 0: CPUPLL

10000030 **CLKCFG1** **Clock Configuration Register 1** **F69F7F00**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_CLK_EN	SDXC_CLK_EN	CRYPTO_CLK_EN	MIPSC_CLK_EN		PCIE_CLK_EN	UPHY_CLK_EN		ETH_CLK_EN			UART2_CLK_EN	UART1_CLK_EN	SPI_CLK_EN	I2S_CLK_EN	I2C_CLK_EN
Type	RW	RW	RW	RW		RW	RW		RW			RW	RW	RW	RW	RW
Reset	1	1	1	1		1	1		1			1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name		GDMA_CLK_EN	PIO_CLK_EN	UART0_CLK_EN	PCM_CLK_EN	MC_CLK_EN	INT_CLK_EN	TIMER_CLK_EN								
Type		RW	RW	RW	RW	RW	RW	RW								
Reset		1	1	1	1	1	1	1								

Bit(s)	Name	Description
31	PWM_CLK_EN	**PWM clock control** 1: Clock Enable 0: Clock Disable
30	SDXC_CLK_EN	**SDXC clock control** 1: Clock Enable 0: Clock Disable
29	CRYPTO_CLK_EN	**AUX system tick counter clock control** 1: Clock Enable 0: Clock Disable
28	MIPSC_CLK_EN	**MIPS Counter clock control** 1: Clock Enable 0: Clock Disable
26	PCIE_CLK_EN	**PCIE2 clock control** 1: Clock Enable 0: Clock Disable
25	UPHY_CLK_EN	**UPHY clock control** 1: Clock Enable 0: Clock Disable
23	ETH_CLK_EN	**ETH clock control** 1: Clock Enable 0: Clock Disable
20	UART2_CLK_EN	**UART2 clock control** 1: Clock Enable 0: Clock Disable
19	UART1_CLK_EN	**UART1 clock control** 1: Clock Enable 0: Clock Disable
18	SPI_CLK_EN	**SPI clock control** 1: Clock Enable 0: Clock Disable
17	I2S_CLK_EN	**I2S clock control** 1: Clock Enable 0: Clock Disable
16	I2C_CLK_EN	**I2C clock control** 1: Clock Enable 0: Clock Disable
14	GDMA_CLK_EN	**GDMA clock control** 1: Clock Enable 0: Clock Disable
13	PIO_CLK_EN	**PIO clock control** 1: Clock Enable 0: Clock Disable
12	UART0_CLK_EN	**UART0 clock control** 1: Clock Enable 0: Clock Disable
11	PCM_CLK_EN	**PCM clock control**

Bit(s)	Name	Description
		1: Clock Enable 0: Clock Disable
10	MC_CLK_EN	**MC clock control** 1: Clock Enable 0: Clock Disable
9	INT_CLK_EN	**INT clock control** 1: Clock Enable 0: Clock Disable
8	TIMER_CLK_EN	**TIMER clock control** 1: Clock Enable 0: Clock Disable

10000034 **RSTCTL** **Reset Control Register** **04000400**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM_RST	SDXC_RST	CRYPTO_RST	AUX_STCK_RST		PCIE_RST		EPHY_RST	ETH_RST	UHST_RST		UART2_RST	UART1_RST	SPI_RST	I2S_RST	I2C_RST
Type	RW	RW	RW	RW		RW		RW	RW	RW		RW	RW	RW	RW	RW
Reset	0	0	0	0		1		0	0	0		0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GDMA_RST	PIO_RST	UART0_RST	PCM_RST	MC_RST	INT_RST	TIMER_RST			HIF_RST	WIFI_RST	SPIS_RST			SYS_RST
Type		RW	RW	RW	RW	RW	RW	RW			RW	RW	RW			W1C
Reset		0	0	0	0	1	0	0			0	0	0			0

Bit(s)	Name	Description
31	PWM_RST	**PWM reset control** 1: Reset Assert 0: Reset Deassert
30	SDXC_RST	**SDXC reset control** 1: Reset Assert 0: Reset Deassert
29	CRYPTO_RST	**Crypto engine reset control** 1: Reset Assert 0: Reset Deassert
28	AUX_STCK_RST	**AUX system tick counter clock control** 1: Reset Assert 0: Reset Deassert
26	PCIE_RST	**PCIE reset control** 1: Reset Assert 0: Reset Deassert
24	EPHY_RST	**EPHY reset control** 1: Reset Assert 0: Reset Deassert
23	ETH_RST	**ETH reset control** 1: Reset Assert 0: Reset Deassert

Bit(s)	Name	Description
22	UHST_RST	**USB PHY reset control** 1: Reset Assert 0: Reset Deassert
20	UART2_RST	**UART2 reset control** 1: Reset Assert 0: Reset Deassert
19	UART1_RST	**UART1 reset control** 1: Reset Assert 0: Reset Deassert
18	SPI_RST	**SPI reset control** 1: Reset Assert 0: Reset Deassert
17	I2S_RST	**I2S reset control** 1: Reset Assert 0: Reset Deassert
16	I2C_RST	**I2C reset control** 1: Reset Assert 0: Reset Deassert
14	GDMA_RST	**GDMA reset control** 1: Reset Assert 0: Reset Deassert
13	PIO_RST	**PIO reset control** 1: Reset Assert 0: Reset Deassert
12	UART0_RST	**UART0 reset control** 1: Reset Assert 0: Reset Deassert
11	PCM_RST	**PCM reset control** 1: Reset Assert 0: Reset Deassert
10	MC_RST	**MC reset control** 1: Reset Assert 0: Reset Deassert
9	INT_RST	**INT reset control** 1: Reset Assert 0: Reset Deassert
8	TIMER_RST	**TIMER reset control** 1: Reset Assert 0: Reset Deassert
5	HIF_RST	**WIFI HIF reset control** [Note] WPDMA reset control 1: Reset Assert 0: Reset Deassert
4	WIFI_RST	**WIFI reset control** [Note] This bit will reset Andes and initialize XTAL and BBPPLL again, MIPS must carefully use it. 1: Reset Assert 0: Reset Deassert
3	SPIS_RST	**SPI Slave control** 1: Reset Assert 0: Reset Deassert
0	SYS_RST	**Whole System Reset Control** [Note] Except for power-on CR, this bit reset the whole system include itself.

Bit(s)	Name	Description
		1: Whole System Reset 0: NA

10000038 **RSTSTAT** **Reset Status Register** **C0030000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**WDT2SYSRST_EN**	**WDT2RSTO_EN**	**WDTRSTPD**													
Type	RW	RW	RW													
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**WDRST_TON9_EN**	**N9_WDRST_EN**					**N9SYSRST**	**SWSYSRST**	**WDRST**	
Type							RW	RW					W1C	W1C	W1C	
Reset							0	0					0	0	0	

Bit(s)	Name	Description
31	WDT2SYSRST_EN	**WDT reset apply to System Reset** Enables watchdog timeout to trigger a system reset. 1: Enable 0: Disable
30	WDT2RSTO_EN	**WDT reset apply to watch dog reset pin out.** 1: Enable 0: Disable
29:16	WDTRSTPD	**Watchdog Reset Output Low Period** Controls the WDT reset output low period. For example: If the pin share mode was set correctly and WDT2RSTO_EN=1, When WDTRSTPD= 0, you can see duration of 1 usec low on the WDT reset output pin. When WDTRSTPD= 3, you can see duration of 4 usec low on the WDT reset output pin. (unit: 1 usec)
9	WDRST_TON9_EN	**MIPS software reset or watch-dog reset apply to N9 subsys.** When this bit is set, MIPS can reset N9 or N9 is reset when MISP watch-dog reset happen. 0: disable 1: Enable
8	N9_WDRST_EN	**N9 watch-dog reset applies to MIPS subsys.** When N9 WDRST happens, N9 will also reset MIPS system. 0: disable 1: Enable
3	N9SYSRST	**N9 watch-dog reset occurred** This bit will be set if N9 wifisys is reset by its watch-dog mechanism. Writing a '1' will clear this bit. Writing a '0' has not effect. NOTE: This register is reset only by a power on reset. 0: Has no effect. 1: Clears this bit.

Bit(s)	Name	Description
2	SWSYSRST	**Software system reset occurred** This bit will be set if software reset the chip by writing to the RSTSYS bit in RSTCTL. Writing a '1' will clear this bit. Writing a '0' has not effect. NOTE: This register is reset only by a power on reset. 0: Has no effect. 1: Clears this bit.
1	WDRST	**Watchdog reset occurred** This bit will be set if the watchdog timer reset the chip. Writing a '1' will clear this bit. Writing a '0' has not effect. NOTE: This register is reset only by power-on reset. 0: Has no effect. 1: Clears this bit.

1000003C **AGPIO_CFG** **Analog GPIO Configuration** **001F001F**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												**EPHY_GPIO_AIO_EN**				**EPHY_P0_DIS**
Type												RW				RW
Reset												1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**RF_OLT_MODE**			**EINT_SEL**	**WLED_OD_EN**				**REF_CLKO_AIO_EN**	**I2S_CLK_AIO_EN**	**I2S_WS_AIO_EN**	**I2S_SDO_AIO_EN**	**I2S_SDI_AIO_EN**
Type				RW			RW	RW				RW	RW	RW	RW	RW
Reset				0			0	0				1	1	1	1	1

Bit(s)	Name	Description
20:17	EPHY_GPIO_AIO_EN	**EPHY P1 ~ P4 digital PAD selection (P1 ~ P4 Disable)** (note: When any bit of bit[20:17] is set to 1, P1 ~ P4 will be swtiched to digital PADs together.) 0: Analog PAD 1: Digital PAD
16	EPHY_P0_DIS	**EPHY P0 Disable** 0: Enable 1: Disable
12	RF_OLT_MODE	**Enable RF OLT mode** 0: DIsable 1: Enable
9	EINT_SEL	**Andes EINT Source** 0: from W_UTIF 1: from GPIO [23:20]
8	WLED_OD_EN	**WLED Open-Drain** 0: DIsable 1: Open-Drain
4	REF_CLKO_AIO_EN	**REF Clock Output PAD Selection** 0: Analog PAD 1: Digital PAD
3	I2S_CLK_AIO_EN	**I2S Clock PAD Selection** 0: Analog PAD

Bit(s)	Name	Description
		1: Digital PAD
2	I2S_WS_AIO_EN	**I2S WS PAD Selection** 0: Analog PAD 1: Digital PAD
1	I2S_SDO_AIO_EN	**I2S CSDO PAD Selection** 0: Analog PAD 1: Digital PAD
0	I2S_SDI_AIO_EN	**I2S SDI PAD Selection** 0: Analog PAD 1: Digital PAD

10000040 **N9_GPIO_INT** **Andes GPIO Interrupt** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**GPIO_INT[16:16]**
Type																W1C
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO_INT[15:0]**															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
16:0	GPIO_INT	**Andes GPIO INT**

10000044 **N9_GPIO_MASK** **Andes GPIO Mask** **0001FFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																**GPIO_MASK[16:16]**
Type																RW
Reset																1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO_MASK[15:0]**															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
16:0	GPIO_MASK	**Andes GPIO MASK**

10000060 **GPIO1_MODE** **GPIO1 purpose selection** **54050404**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PWM1_MODE		PWM0_MODE		UART2_MODE		UART1_MODE				I2C_MODE			REFCLK_MODE		PERST_MODE
Type	RW		RW		RW		RW				RW			RW		RW
Reset	0	1	0	1	0	1	0	0			0	0		1		1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		WDT_MODE		SPI_MODE	SD_MODE		UART0_MODE		I2S_MODE		SPI_CS1_MODE		SPIS_MODE		GPIO_MODE	
Type		RW		RW	RW		RW		RW		RW		RW		RW	
Reset		0		0	0	1	0	0	0	0	0	0	0	1	0	0

Bit(s)	Name	Description
31:30	PWM1_MODE	**PWM1 GPIO mode** 3: SDXC D6 2: UTIF[5] 1: GPIO 0: PWM ch1
29:28	PWM0_MODE	**PWM0 GPIO mode** 3: SDXC D7 2: UTIF[4] 1: GPIO 0: PWM ch0
27:26	UART2_MODE	**UART2 GPIO mode** 3: SDXC D5/D4 2: PWM ch2/ch3 1: GPIO 0: UART-Lite #2
25:24	UART1_MODE	**UART1 GPIO mode** 3: SW_R, SW_T 2: PWM ch0/ch1 1: GPIO 0: UART-Lite #1
21:20	I2C_MODE	**I2C GPIO mode** 2: S-UART (debug) 1: GPIO 0: I2C
18	REFCLK_MODE	**REFCLK GPIO mode** 1: GPIO 0: REFCLK (12M)
16	PERST_MODE	**PCIe RESET GPIO mode** 1: GPIO 0: PCIe reset
14	WDT_MODE	**Watch dog timeout GPIO mode** 1: GPIO 0: Watch dog
12	SPI_MODE	**SPI GPIO mode** 1: GPIO 0: SPI
11:10	SD_MODE	**SDXC GPIO mode** 3: Andes JTAG 2: UTIF[17:10] 1: GPIO 0: SDXC

Bit(s)	Name	Description
9:8	UART0_MODE	**UART0 GPIO mode** 1: GPIO 0: UART-Lite #0
7:6	I2S_MODE	**I2S GPIO mode** 3: ANTSEL[5:2] 2: PCM 1: GPIO 0: I2S
5:4	SPI_CS1_MODE	**SPI CS1 GPIO mode** 2: REFCLK 1: GPIO 0: SPI CS1
3:2	SPIS_MODE	**SPI Slave GPIO mode** 3: PWM CH0/1 and UART2 2: UTIF[3:0] 1: GPIO 0: SPI Slave
1:0	GPIO_MODE	**GPIO mode** 3: PCIe Reset 2: REFCLK (12M) 1: GPIO 0: GPIO

10000064 GPIO2_MODE GPIO2 purpose selection 05550555

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**P4_LED_KN_MODE**		**P3_LED_KN_MODE**		**P2_LED_KN_MODE**		**P1_LED_KN_MODE**		**P0_LED_KN_MODE**		**WLED_KN_MODE**	
Type					RW		RW		RW		RW		RW		RW	
Reset					0	1	0	1	0	1	0	1	0	1	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**P4_LED_AN_MODE**		**P3_LED_AN_MODE**		**P2_LED_AN_MODE**		**P1_LED_AN_MODE**		**P0_LED_AN_MODE**		**WLED_AN_MODE**	
Type					RW		RW		RW		RW		RW		RW	
Reset					0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
27:26	P4_LED_KN_MODE	**EPHY P4 LED GPIO mode** [Note] Only valid for MT7628KN. 3: JTAG (JTRST_N) 2: UTIF[6] 1: GPIO 0: EPHY P4 LED
25:24	P3_LED_KN_MODE	**EPHY P3 LED GPIO mode** [Note] Only valid for MT7628KN. 3: JTAG (JTCLK) 2: UTIF[7] 1: GPIO 0: EPHY P3 LED
23:22	P2_LED_KN_MODE	**EPHY P2 LED GPIO mode** [Note] Only valid for MT7628KN. 3: JTAG (JTMS) 2: UTIF[8] 1: GPIO 0: EPHY P2 LED

Bit(s)	Name	Description
21:20	P1_LED_KN_MODE	**EPHY P1 LED GPIO mode** [Note] Only valid for MT7628KN. 3: JTAG (JTDI) 2: UTIF[9] 1: GPIO 0: EPHY P1 LED
19:18	P0_LED_KN_MODE	**EPHY P0 LED GPIO mode** [Note] Only valid for MT7628KN. 3: JTAG(JTDO) 2: Reserved 1: GPIO 0: EPHY P0 LED
17:16	WLED_KN_MODE	**WLED GPIO mode** [Note] Only valid for MT7628KN. 3: Reserved 2: Reserved 1: GPIO 0: WLED
11:10	P4_LED_AN_MODE	**EPHY P4 LED GPIO mode** [Note] Only valid for MT7628AN. 3: JTAG (JTRST_N) 2: UTIF[6] 1: GPIO 0: EPHY P4 LED
9:8	P3_LED_AN_MODE	**EPHY P3 LED GPIO mode** [Note] Only valid for MT7628AN. 3: JTAG (JTCLK) 2: UTIF[7] 1: GPIO 0: EPHY P3 LED
7:6	P2_LED_AN_MODE	**EPHY P2 LED GPIO mode** [Note] Only valid for MT7628AN. 3: JTAG (JTMS) 2: UTIF[8] 1: GPIO 0: EPHY P2 LED
5:4	P1_LED_AN_MODE	**EPHY P1 LED GPIO mode** [Note] Only valid for MT7628AN. 3: JTAG (JTDI) 2: UTIF[9] 1: GPIO 0: EPHY P1 LED
3:2	P0_LED_AN_MODE	**EPHY P0 LED GPIO mode** [Note] Only valid for MT7628AN. 3: JTAG(JTDO) 2: Reserved 1: GPIO 0: EPHY P0 LED
1:0	WLED_AN_MODE	**WLED GPIO mode** [Note] Only valid for MT7628AN. 3: Reserved 2: Reserved 1: GPIO 0: WLED

10000068 **MEMO1** **Memory1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMO1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MEMO1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MEMO1	**Memory1**

1000006C **MEMO2** **Memory2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMO2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MEMO2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MEMO2	**Memory2**

10000070 **EXT_MEMO1** **Extend Application #1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMO1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MEMO1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MEMO1	**Extend Application #1**

10000074 **EXT_MEMO2** **Extend Application #2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMO2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MEMO2[15:0]															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MEMO2	**Extend Application #2**

10000078 EXT_MEMO3 Extend Application #3 **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMO3[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MEMO3[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MEMO3	**Extend Application #3**

1000007C EXT_MEMO4 Extend Application #4 **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMO4[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MEMO4[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MEMO4	**Extend Application #4**

2.3 Timer

2.3.1 Features

- Independent 1usec tick pre-scale for each timer.
- Independent interrupts for each timer.
- Two general-purpose timers and a watchdog timer. Watchdog timer resets system on time-out.
- Timer Modes
  - *Periodic*
    In periodic mode, the timer counts down to zero from the limited value. An interrupt is generated when the count is zero. After reaching zero, the limited value is reloaded into the timer and the timer counts down again. A limited value of zero disables the timer.
  - *Timeout*
    In timeout mode, the timer counts down to zero from the limited value. An interrupt is generated when the count is zero. In this mode, the ENABLE bit is reset when the timer reaches zero, stopping the counter.
  - *Watchdog*
    In watchdog mode, the timer counts down to zero from the limited value. If the load value is not reloaded or the timer is not disabled before the count is zero, the chip will be reset. When this occurs, every register in the chip is reset except the watchdog reset status bit WDRST in the RSTSTAT register in the system control block; it remains set to alert firmware of the timeout event when it re-executes its bootstrap.

2.3.2 Block Diagram

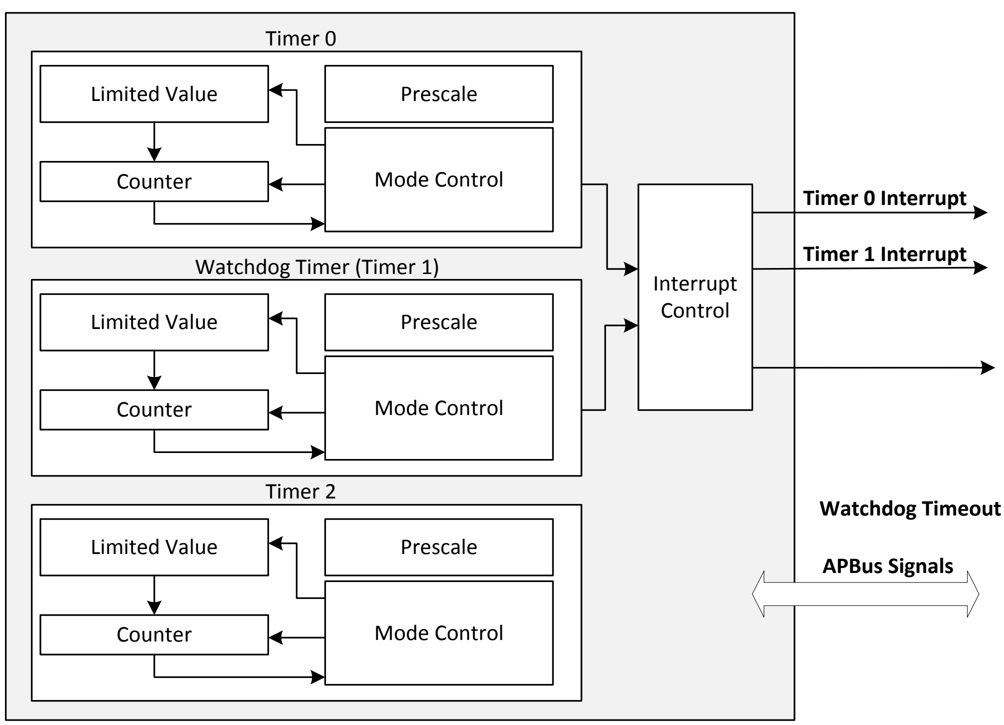


Figure 2-2 Timer Block Diagram

2.3.3 Registers

TIMER Changes LOG

Revision	Date	Author	Change Log
0.1	2012/8/24	Leon Chung	Initialization
0.2	2013/12/10	Rick Ho	1. Modify T0CTL_REG Bit[4] to WO and add Bit[3] RO 2. Modify WDTCTL_REG Bit[4] to WO and add Bit[3] RO 3. Modify T1CTL_REG Bit[4] to WO and add Bit[3] RO

Module name: TIMER Base address: (+10000100h)

Address	Name	Width	Register Function
10000100	TGLB_REG	32	**RISC Global Control Register**
10000110	T0CTL_REG	32	**RISC Timer 0 Control Register**
10000114	T0LMT_REG	32	**RISC Timer 0 Limit Register**
10000118	T0_REG	32	**RISC Timer 0 Register**
10000120	WDTCTL_REG	32	**Watch Dog Timer Control Register**
10000124	WDTLMT_REG	32	**Watch Dog Timer Limit Register**
10000128	WDT_REG	32	**Watch Dog Timer Register**
10000130	T1CTL_REG	32	**RISC Timer 1 Control Register**
10000134	T1LMT_REG	32	**RISC Timer 1 Limit Register**
10000138	T1_REG	32	**RISC Timer 1 Register**

10000100 TGLB_REG RISC Global Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV1[20:5]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV1[4:0]					T1RST	WDTRST	T0RST	RESV0					T1INT	WDTINT	T0INT
Type	RO					W1C	W1C	W1C	RO					W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:11	RESV1	**Reserved**
10	T1RST	**Timer 1 reset** 1: to reset timer 1 to T1LMT value
9	WDTRST	**Watch dog timer reset** 1: to reset watch dog timer to WDTLMT value
8	T0RST	**Timer 0 reset** 1: to reset timer 0 to T0LMT value
7:3	RESV0	**Reserved**
2	T1INT	**Timer 1 interrupt status**
1	WDTINT	**Watch dog timer interrupt status**

Bit(s)	Name	Description
0	T0INT	**Timer 0 interrupt status**

10000110 **T0CTL_REG** **RISC Timer 0 Control Register** **0000000 0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**T0PRES**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESV2**								**T0EN**	**RESV1**		**T0AL**	**T0AL_STATUS**	**RESV0**		
Type	RO								RW	RO		WO	RO	RO		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	T0PRES	**Timer 0 count down tick pre-scale. Unit is 1u second.**
15:8	RESV2	**Reserved**
7	T0EN	**Timer 0 count down enable**
6:5	RESV1	**Reserved**
4	T0AL	**Timer 0 auto load enable** 1: Enable 0: Disable
3	T0AL_STATUS	**Timer 0 auto load enable status** 1: Enable 0: Disable
2:0	RESV0	**Reserved**

10000114 **T0LMT_REG** **RISC Timer 0 Limit Register** **0000000 0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV0**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**T0LMT**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV0	**Reserved**
15:0	T0LMT	**Timer 0 Limit.** When T0AL is set to 1, T0LMT will be loaded into timer 0 when timer 0 is enabled or when count down to 0.

10000118 **T0_REG** **RISC Timer 0 Register** **0000FFF F**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	T0															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:16	RESV0	**Reserved**
15:0	T0	**RISC down-count timer 0**

10000120 **WDTCTL_REG** **Watch Dog Timer Control Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WDTPRES															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV2								WDTEN	RESV1		WDTAL	WDTAL_STATUS	RESV0		
Type	RO								RW	RO		WO	RO	RO		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	WDTPRES	**Watch dog timer count down tick pre-scale. Unit is 1u second.**
15:8	RESV2	**Reserved**
7	WDTEN	**Watch dog timer count down enable**
6:5	RESV1	**Reserved**
4	WDTAL	**Watch dog timer auto load enable** 1: Enable 0: Disable
3	WDTAL_STATUS	**Watch dog timer auto load enable status** 1: Enable 0: Disable
2:0	RESV0	**Reserved**

10000124 **WDTLMT_REG** **Watch Dog Timer Limit Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WDTLMT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:16	RESV0	**Reserved**
15:0	WDTLMT	**Watch dog timer Limit.** When WDTAL is set to 1, WDTLMT will be loaded into watch dog timer when watch dog timer is enabled or when count down to 0.

10000128 **WDT_REG** **Watch Dog Timer Register** **0000FFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WDT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:16	RESV0	**Reserved**
15:0	WDT	**watch dog timer.**

10000130 **T1CTL_REG** **RISC Timer 1 Control Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	T1PRES															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV2								T1EN	RESV1		T1AL	T1AL_STATUS	RESV0		
Type	RO								RW	RO		WO	RO	RO		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	T1PRES	**Timer 1 count down tick pre-scale. Unit is 1u second.**
15:8	RESV2	**Reserved**
7	T1EN	**Timer 1 count down enable**
6:5	RESV1	**Reserved**
4	T1AL	**Timer 1 auto load enable** 1: Enable 0: Disable
3	T1AL_STATUS	**Timer 1 auto load enable status** 1: Enable 0: Disable
2:0	RESV0	**Reserved**

10000134 **T1LMT_REG** **RISC Timer 1 Limit Register** **0000000**

0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV0**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**T1LMT**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV0	**Reserved**
15:0	T1LMT	**Timer 1 Limit.** When T1AL is set to 1, T1LMT will be loaded into timer 1 when timer 1 is enabled or when count down to 0.

10000138 **T1_REG** **RISC Timer 1 Register** **0000FFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV0**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**T1**															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:16	RESV0	**Reserved**
15:0	T1	**RISC down-count timer 1**

2.4 Interrupt Controller

2.4.1 Registers

CIRQ Changes LOG

Revision	Date	Author	Change Log
0.1	2012/6/15	YuShu Xiao	Initialization

Module name: CIRQ Base address: (+10000200h)

Address	Name	Width	Register Function
10000200	IRQ_SEL0	32	**IRQ Selection 0 Register** The registers allow the interrupt sources to be mapped onto interrupt requests IRQ. When write data to this register, the FIQ_SEL register will be update to the inverse data at the same time.
10000204	IRQ_SEL1	32	**Reserved** Reserved
10000208	IRQ_SEL2	32	**Reserved** Reserved
1000020C	IRQ_SEL3	32	**Reserved** Reserved
1000026C	FIQ_SEL	32	**FIQ Selection Register** The registers allow the interrupt sources to be mapped onto interrupt requests FIQ. When write data to this register, the IRQ_SEL0 register will be update to the inverse data at the same time.
10000270	IRQ_MASK	32	**IRQ Mask Register** This register contains a mask bit for each interrupt line in IRQ Controller.
10000274	FIQ_MASK	32	**FIQ Mask Register** This register contains a mask bit for each interrupt line in FIQ Controller
10000278	IRQ_MASK_CLR	32	**IRQ Mask Clear Register** This register is used to clear bits in IRQ Mask Register.
1000027C	FIQ_MASK_CLR	32	**FIQ Mask Clear Register** This register is used to clear bits in FIQ Mask Register.
10000280	IRQ_MASK_SET	32	**IRQ Mask Set Register** This register is used to set bits in the IRQ Mask Register.
10000284	FIQ_MASK_SET	32	**FIQ Mask Set Register** This register is used to set bits in the FIQ Mask Register.
10000288	IRQ_EOI	32	**IRQ End of Interrupt Register** This register provides a mean for software to relinquish and to refresh the interrupt controller. Writing a 1 to a specific bit results in an IRQ End of Interrupt command issued internally to the corresponding interrupt line.
1000028C	FIQ_EOI	32	**FIQ End of Interrupt Register** This register provides a mean for software to relinquish and to refresh the interrupt controller. Writing a 1 to a specific bit results in an FIQ End of Interrupt command issued internally to the corresponding interrupt line.
10000290	IRQ_SENS	32	**IRQ Sensitive Register** This register is used to set the IRQ interrupts as either edge or

			level sensitive.
10000294	FIQ_SENS	32	**FIQ Sensitive Register** This register is used to set the FIQ interrupts as either edge or level sensitive.
10000298	INT_SOFT	32	**Software Interrupt Register** Setting 1 to the specific bit position generates a software interrupt for corresponding interrupt line before interrupt input multiplex. This register is used for debug purpose.
1000029C	IRQ_STAT	32	**IRQ Status Register** Reading this register will get the IRQ interrupt sources with masking.
100002A0	FIQ_STAT	32	**FIQ Status Register** Reading this register will get the FIQ interrupt sources with masking.
100002A4	INT_PURE	32	**Interrupt Pure Register** Reading this register will get the pure interrupt sources without masking.
100002A8	INT_MSEL	32	**Interrupt Mode Selection Register** This register is used to select the interrupt modes of MIPS1004Kc.

10000200 IRQ_SEL0 IRQ Selection 0 Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:0	IRQ0	**IRQ Selection 0** 0: Clear IRQ_SEL0 and Set FIQ_SEL 1: Set IRQ_SEL0 and Clear FIQ_SEL

10000204 IRQ_SEL1 Reserved 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RESV	**Reserved**

10000208 **IRQ_SEL2** **Reserved** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RESV	**Reserved**

1000020C **IRQ_SEL3** **Reserved** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RESV	**Reserved**

1000026C **FIQ_SEL** **FIQ Selection Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	FIQ	**FIQ Selection** 0: Clear FIQ_SEL and Set IRQ_SEL0 1: Set FIQ_SEL and Clear IRQ_SEL0

10000270 **IRQ_MASK** **IRQ Mask Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	IRQ0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	IRQ0	**IRQ Mask** 0: Interrupt is disabled 1: Interrupt is enabled

10000274 **FIQ_MASK** **FIQ Mask Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	FIQ	**FIQ Mask** 0: Interrupt is disabled 1: Interrupt is enabled

10000278 **IRQ_MASK_CLR** **IRQ Mask Clear Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ0[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	IRQ0	**IRQ Mask Clear** 0: No effect 1: Clear the corresponding MASK bit

1000027C **FIQ_MASK_CLR** **FIQ Mask Clear Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	FIQ	**FIQ Mask Clear** 0: No effect 1: Clear the corresponding MASK bit

10000280 **IRQ_MASK_SET** **IRQ Mask Set Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ0[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	IRQ0	**IRQ Mask Set** 0: No effect 1: Set the corresponding MASK bit

10000284 **FIQ_MASK_SET** **FIQ Mask Set Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	FIQ	**FIQ Mask Set** 0: No effect 1: Set the corresponding MASK bit

10000288 **IRQ_EOI** **IRQ End of Interrupt Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ0[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	IRQ0	**IRQ End of Interrupt** 0: No service is currently in progress or pending

Bit(s)	Name	Description
		1: Interrupt request is in-service

1000028C FIQ_EOI FIQ End of Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	FIQ	**FIQ End of Interrupt** 0: No service is currently in progress or pending 1: Interrupt request is in-service

10000290 IRQ_SENS IRQ Sensitive Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:0	IRQ0	**IRQ Sensitive** 0: Edge sensitivity with Pos-edge Edge 1: Level sensitivity with active High

10000294 FIQ_SENS FIQ Sensitive Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:0	FIQ	**FIQ Sensitive** 0: Edge sensitivity with Pos-edge Edge 1: Level sensitivity with active High

10000298 **INT_SOFT** **Software Interrupt Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INT[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INT[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	INT	**Software Interrupt**

1000029C **IRQ_STAT** **IRQ Status Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQ0[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQ0[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	IRQ0	**IRQ Status** 0: No interrupt request is generated 1: Interrupt request is pending

100002A0 **FIQ_STAT** **FIQ Status Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIQ[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FIQ[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	FIQ	**FIQ Status** 0: No interrupt request is generated 1: Interrupt request is pending

100002A4 **INT_PURE** **Interrupt Pure Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	INT[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INT[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	INT	**Pure Interrupt** 0: No interrupt source is asserted 1: Interrupt source is asserted

100002A8 **INT_MSEL** **Interrupt Mode Selection Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[30:15]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[14:0]															SEL
Type	RW															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:1	RESV	**Reserved**
0	SEL	**Interrupt Mode Selection** 0: Compatibility & Vectored Interrupt Mode 1: External Interrupt Controller Mode

2.5 EMC Controller

2.5.1 Regsiter

EXT_MC_ARB Changes LOG

Revision	Date	Author	Change Log
0.1	2012/10/5	Lancelot	Initialization
0.2	2013/8/19	YS Xiao	Modify to MT7628

Module name: EXT_MC_ARB Base address: (+10000300h)

Address	Name	Width	Register Function
10000300	SDRAM_CFG0	32	SDRAM Configuration 0
10000304	SDRAM_CFG1	32	SDRAM Configuration 1
10000308	ILL_ACC_ADDR	32	Illegal Access Address Capture
1000030C	ILL_ACC_TYPE	32	Illegal Access Type Capture
10000310	DDR_SELF_REFRESH	32	ODT and Self-Refresh Configuration
10000314	SDR_DDR_PWR_SAVE_CNT	32	Self-Refresh Time Count
10000320	DLL_DBG	32	DRAM DLL Debug Probe
10000340	DDR_CFG0	32	DDR1/DDR2 controller configuration 0 register
10000344	DDR_CFG1	32	DDR1/DDR2 controller configuration 1 register
10000348	DDR_CFG2	32	DDR1/DDR2 controller configuration 2 register
1000034C	DDR_CFG3	32	DDR1/DDR2 controller configuration 3 register
10000350	DDR_CFG4	32	DDR1/DDR2 controller configuration 4 register
10000360	DDR_DQ_DLY	32	DDR1/DDR2 DQ delay control register
10000364	DDR_DQS_DLY	32	DDR1/DDR2 DQS delay control register
10000368	DDR_DLL_SLV	32	DDR1/DDR2 DLL slave control register
1000036C	DDR_DLL_MST	32	DDR1/DDR2 DLL master control register
10000380	MC_ARB_CFG	32	MC 2 to 1 arbiter setting
10000384	MC_AG_BW	32	MC Channel BW/QoS_Type/DueDate Setting
10000390	RB_DBG	32	RB Debug
10000394	RB_STATE	32	RB Debug State
10000398	RB_BW	32	RB Bandwidth
1000039C	RB_LAT	32	RB Latency

10000300 SDRAM_CFG0 SDRAM Configuration 0 — 5192528 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIS_CLK_GT	CLK_SLEW		TWR	TMRD				TRFC				RSV0		TCAS	
Type	RW	RW		RW	RW				RW				RO		RW	
Reset	0	1	0	1	0	0	0	1	1	0	0	1	0	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TRAS				RSV1		TRCD		TRC				RSV2		TRP	

Type	RW				RO		RW		RW				RO		RW	
Reset	0	1	0	1	0	0	1	0	1	0	0	0	0	0	1	0

Bit(s)	Name	Description
31	DIS_CLK_GT	**Disable Clock Gating** Disables clock gating of the SDR DRAM controller. 0: Enable 1: Disable
30:29	CLK_SLEW	**Reserved**
28	TWR	**Write Recovery Time** (unit: system clock cycles - 1)
27:24	TMRD	**Load Mode Register command to any other command delay.** (unit: system clock cycles - 1)
23:20	TRFC	**Auto Refresh period** (unit: system clock cycles - 1)
19:18	RSV0	**Reserved**
17:16	TCAS	**CAS Latency Time** (unit: system clock cycles - 1)
15:12	TRAS	**The Active To Precharge command delay.** (unit: system clock cycles - 1)
11:10	RSV1	**Reserved**
9:8	TRCD	**Active To Read or Write delay (RAS to CAS delay)** (unit: system clock cycles - 1)
7:4	TRC	**Active To Active command period** (unit: system clock cycles - 1)
3:2	RSV2	**Reserved**
1:0	TRP	**Precharge command period** (unit: system clock cycles - 1)

10000304 **SDRAM_CFG1** **SDRAM Configuration 1** **01120600**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SDRAM_INIT_START	SDRAM_INIT_DONE	RBC_MAPPING	PWR_DOWN_EN	PWR_DOWN_MODE	RSV0		SDRAM_WIDTH	RSV1		NUMCOLS		RSV2		NUMROWS	
Type	RW	RO	RW	RW	RW	RO		RW	RO		RW		RO		RW	
Reset	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TREFR															
Type	RW															
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	SDRAM_INIT_START	**SDRAM Initialization Start performs the SDRAM initialization sequence.** Can not set this bit to 0 after initialization. 1: Start intialization
30	SDRAM_INIT_DONE	**SDRAM Initialization Done** Indicates the SDRAM has been initialized. 0: Not initialized.

Bit(s)	Name	Description
		1: Initialized.
29	RBC_MAPPING	**RBC Mapping** Selects the address mapping scheme. 0: {BANK ADDR, ROW ADDR, COL ADDR} address mapping scheme 1: {ROW ADDR, BANK ADDR, COL ADDR} address mapping scheme
28	PWR_DOWN_EN	**Power Down Enable** Enables the SDRAM precharge power-down mode to save standby power. 0: Disable 1: Enable
27	PWR_DOWN_MODE	**Power Down Mode** 0: Precharge power down mode 1: Active power down
26:25	RSV0	**Reserved**
24	SDRAM_WIDTH	**SDRAM Width** Selects the number of SDRAM data bus bits. 0: 16 bits 1: 32 bits
23:22	RSV1	**Reserved**
21:20	NUMCOLS	**Number of Columns** Selects the number of column address bits. 0: 8 Column address bits 1: 9 Column address bits (default) 2: 10 Column address bits 3 11 Column address bits
19:18	RSV2	**Reserved**
17:16	NUMROWS	**Number of Rows** Selects the number of row address bits. 0: 11 Row address bits 1: 12 Row address bits (default) 2: 13 Row address bits 3: 14 Row address bits
15:0	TREFR	**AUTO REFRESH period** (unit: SDRAM clock cycles - 1).

10000308 **ILL_ACC_ADDR** **Illegal Access Address Capture** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ILL_ACC_ADDR[31:16]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ILL_ACC_ADDR[15:0]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	ILL_ACC_ADDR	**Illegal Access Address if any bus masters (including CPU) issue illegal accesses (e.g. accesses to reserved memory space, or non-double-word accesses to configuration registers), the address of the illegal transaction is captured in this register.** An illegal interrupt is generated to indicate this exception.

1000030C **ILL_ACC_TYPE** **Illegal Access Type Capture** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ILL_INT_STATUS	ILL_ACC_WR	RSV0										ILL_ACC_BSEL			
Type	W1C	RO	RO										RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1					ILL_IID			ILL_ACC_LEN							
Type	RO					RO			RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	ILL_INT_STATUS	**Illegal Access Interrupt Status** Indicates whether the ilegal access interrupt is cleared or pending. Read 0: Cleared 1: Pending Write 1: Clear both the ILL_ACC_ADDR and ILL_ACC_TYPE registers and thus clear ILL_INT_STATUS.
30	ILL_ACC_WR	**Illegal Access Write** Indicates the illegal access is a read or a write. 0: A read access 1: A write access
29:20	RSV0	**Reserved**
19:16	ILL_ACC_BSEL	**Illegal Access Byte Select** Indicates which bytes were illegally accessed.
15:11	RSV1	**Reserved**
10:8	ILL_IID	**Illegal Access Initiator ID** Indicates the initiator ID of the illegal access. 0: CPU 1: DMA 2: PPE 3: Ethernet PDMA Rx 4: Ethernet PDMA Tx 5: PCI/PCIE 6: Embedded WLAN MAC/BBP 7: USB
7:0	ILL_ACC_LEN	**Illegal Access Length** Indicates the access size of the illegal access. (unit: bytes)

10000310 **DDR_SELF_REFRESH** **ODT and Self-Refresh Configuration** **0E120001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0				ODT_SRC_SEL				ODT_OFF_DLY				ODT_ON_DLY			
Type	RO				RW				RW				RW			
Reset	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1											SR_AU	RSV2		SRAC	SRRE

												TO_EN			K_B	Q_B
Type	RO											RW	RO		RO	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:28	RSV0	**Reserved**
27:24	ODT_SRC_SEL	**ODT Source Select** Sets the DDR pad ODT control source. 0: Dasavtive[0] 1: Dasavtive[1] ... 11: Dasavtive[11] 12: DQS_WINDOW 13: ODT_LOCAL 14: Always on 15: Always off
23:20	ODT_OFF_DLY	**ODT Off Delay** Sets the delay time of the ODT_OFF signal based on the ODT_ON signal. 0: 0 T 1: 0.5 T 2: 1.5 T 3: 2.5 T ... 15: 14.5 T
19:16	ODT_ON_DLY	**ODT On Delay** Sets the delay time of the ODT_ON signal based on the ODT source signal. 0: 0 T 1: 1 T 2: 2 T ... 15: 15 T
15:5	RSV1	**Reserved**
4	SR_AUTO_EN	**Auto Self-Refresh Enable** Enables auto self-refresh for power saving. 0: Disable 1: Enable
3:2	RSV2	**Reserved**
1	SRACK_B	**Self-Refresh Acknowledge Status** Indicates whether DDR2 is in self-refresh mode or has exited from self-refresh mode. When DDR2 changes from self-refresh mode to normal mode, it takes about 200 clock cycles. 0: The DDR2 is in self-refresh mode. 1: The DDR2 has exited from self-refresh mode.
0	SRREQ_B	**Self-Refresh Request Control** Requests DDR2 to enter or exit self-refresh mode. It is low active. 0: Enter self-refresh mode. 1: Exit self-refresh mode.

10000314	SDR_DDR_PWR_SAVE_CNT	Self-Refresh Time Count	0003FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PD_CNT								SR_TAR_CNT[23:16]							

Type	RO								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SR_TAR_CNT[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:24	PD_CNT	**Power Down Count** Counts the times self-refresh mode is entered
23:0	SR_TAR_CNT	**Self-Refresh Time Count** This counter is only referenced when the SDR (PWR_DOWN_EN) or DDR (SR_AUTO_EN) is set. This counter measures the period SDR or DDR is in IDLE status. When the IDLE period has reached the specified time period, the SDR or DDR automatically enter power-saving or selfrefresh mode. Use the following equations to configure the counter. DRAM_CLK_FREQ is PLL_CLK (600 MHz) divided by 3 DDR: (SR_TAR_CNT * 256 + 255) / DRAM_CLK_FREQ SDR: (SR_TAR_CNT * 256) / DRAM_CLK_FREQ DDR reference table 200 MHz: (32'h03FFFF * 256 + 255) * 5 ns ~= 335 ms SDRAM reference table 120 MHz: 32'h03FFFF * 256 * 8.3 ns ~= 560 ms

10000320 **DLL_DBG** **DRAM DLL Debug Probe** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0												RSV1		TDC_STABLE[5:4]	
Type	RO												RO		RO	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TDC_STABLE[3:0]				MST_DLY_SEL								RSV2	CURR_STATE		ADLL_LOCK_DONE
Type	RO				RO								RO	RO		RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:20	RSV0	**Reserved**
19:18	RSV1	**Reserved**
17:12	TDC_STABLE	**ADLL master coarse-grain delay code**
11:4	MST_DLY_SEL	**ADLL master final delay code**
3	RSV2	**Reserved**
2:1	CURR_STATE	**ADLL controller FSM current state**
0	ADLL_LOCK_DONE	**ADLL lock done signal**

10000340 **DDR_CFG0** **DDR1/DDR2 controller configuration 0 register** **249B425B**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**T_RRD**				**T_RAS**					**T_RP**				**T_RFC[5:3]**		
Type	RW				RW					RW				RW		
Reset	0	0	1	0	0	1	0	0	1	0	0	1	1	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**T_RFC[2:0]**			**T_REFI**												
Type	RW			RW												
Reset	0	1	0	0	0	0	1	0	0	1	0	1	1	0	1	1

Bit(s)	Name	Description
31:28	T_RRD	**The minimum number of clock cycles from an active command to the next active command for different banks (TRRD). For DDR2 devices, this is required to be a minimum of 2 regardless of the cycle time.**
27:23	T_RAS	**The number of clock cycles from an active command until a pre-charge command is allowed. To obtain this value, one should divide the minimum RAS# to pre-charge delay of the SDRAM by the clock cycle time (TRAS). The sum of Active-to-Pre-charge and Pre-charge-to-Active should be equal or larger than active-to-active delay of the same ban (TRC)**
22:19	T_RP	**The number of clock cycles needed for the SDRAM to recover from a pre-charge command and ready to accept the next active command. To obtain this value, one should divide the RAS# pre-charge time of the SDRAM (TRP) by the clock cycle time. The sum of Active-to-Pre-charge and Pre-charge-to-Active should be equal or larger than active-to-active delay of the same bank (TRC)**
18:13	T_RFC	**Half the number of clock cycles needed for the SDRAM to recover from a refresh signal to be ready to take the next command. To obtain this value, one should divide the SDRAM row cycle time (TRFC) by the clock cycle time.**
12:0	T_REFI	**The number of clock cycles from one refresh command to the next refresh command. To obtain this value, one should divide the periodic refresh interval (TREFI) by the clock cycle time. The actual timing of issuing a pre-charge command may be delayed by if the SDRAM is processing a normal access. However, the delay is not accumulative so there is no need to shorten the refresh interval to account for memory access time. The non-accumulative refresh delay typically increases memory bandwidth by a few percentage points.**

10000344 **<u>DDR_CFG1</u>** **DDR1/DDR2 controller configuration 1 register** **222E2424**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**T_WTR**				**T_RTP**				**RSV0**		**USER_DATA_WIDTH**	**IND_SDRAM_SIZE**			**IND_SDRAM_WIDTH**	
Type	RW				RW				RO		RW	RW			RW	
Reset	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EXT_BANK**		**TOTAL_SDRAM_WIDTH**		**T_WR**				**T_MRD**				**T_RCD**			
Type	RW		RW		RW				RW				RW			
Reset	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0

Bit(s)	Name	Description
31:28	T_WTR	**The write-to-read delay (TWTR) (last write data to the next read**

Bit(s)	Name	Description
		command) as specified by the DDR2 data sheet
27:24	T_RTP	**The read-to-pre-charge delay (TRTP) as specified by the DDR2 data sheet. Note that this is a DDR2 requirement, and requires a minimum of 2 cycles. These bits are ignored in DDR mode.**
23:22	RSV0	**Reserved**
21	USER_DATA_WIDTH	**Specify user data width** 0: 32-bit 1: 64-bit When user data width is 32-bit, total SDRAM width (bit[13:12]) must be 10. NOTE: This system is always 64-bit. Please do not modify this setting.
20:18	IND_SDRAM_SIZE	**Specify individual SRAM size** 000: Reserved 001: Individual SDRAM is 64 Mbit, (DDR only) 010: Individual SDRAM is 128 Mbit, (DDR only) 011: Individual SDRAM is 256 Mbit. 100: Individual SDRAM is 512 Mbit. 101: Individual SDRAM is 1 Gbit. 110: Individual SDRAM is 2 Gbit, (DDR2 only). 111: Reserved
17:16	IND_SDRAM_WIDTH	**Specify individual SRAM data width** 00: Reserved 01: 8-bit. 10: 16-bit. 11: Reserved
15:14	EXT_BANK	**Specify bank/module configuration** 00: 1 external bank, 1 module. (CS#[0]) 01: 2 external bank, 1 module. (CS#[1:0]), 10: Reserved 11: 2 external banks, 2 modules. (CS#[1:0]) NOTE: only one CS pin.
13:12	TOTAL_SDRAM_WIDTH	**This field specifies the total data width to the SDRAM. For example, if four 8-bit wide DDR2 chips are used in parallel to form a 32-bit DDR2 data width, this field should be defined as 11 to indicate a 32-bit width. In this case, bit[17:16] should be defined as 01.** 00: Reserved 01: Reserved 10: 16-bit 11: 32-bit. Allowed only when user data width is 64-bit (bit21 is 1).
11:8	T_WR	**The clock cycles needed for the DDR to recover from a write command and be able to accept a pre-charge command. To obtain this value, divide the SDRAM write recovery time by the clock cycle time (TWR)**
7:4	T_MRD	**The number of clock cycles after the setting of the mode registers in the DDR and before the issue of the next command. To obtain this value, divide the Mode Register Set Cycle time (TMRD) by the clock cycle time.**
3:0	T_RCD	**The number of clock cycles from an active command to a read/write assertion. To obtain this value, divide the RAS# to CAS# delay time (TRCD) by the clock cycle time.**

10000348 DDR_CFG2 DDR1/DDR2 controller configuration 2 register 43FFE443

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REGE	DDR2_MODE	DQS0_GATING_WINDOW		DQS1_GATING_WINDOW		RSV0[12:3]									

Type	RW	RW	RW		RW		RO									
Reset	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[2:0]			PD	WR			DLL RE SET	TES TM OD E	CAS_LATENCY			BU RS T_T YP E	BURST_LENGTH		
Type	RO			RW	RW			RW	RW	RW			RO	RW		
Reset	1	1	1	0	0	1	0	0	0	1	0	0	0	0	1	1

Bit(s)	Name	Description
31	REGE	**This bit should be high when external registers are inserted in the controller and address signals are sent between the controller and the DDR SDRAM. One example of such instance is when register mode SDRAM DIMM is used. This bit should be low when the control and address signals from the controller is connected to the SDRAM without register delay.**
30	DDR2_MODE	**This bit determines whether the memory controller is in DDR1 or DDR2 mode.** 0: DDR1 mode 1: DDR2 mode
29:28	DQS0_GATING_WINDOW	**Controls the mask for the data strobe 0 (DQS0) window leading and trailing edge.** 00: Half extended cycle for the leading and trailing edge of DQS window (maximum window) 01: Only half extended cycle for leading edge of DQS window 10: Only half extended cycle for trailing edge of DQS window 11: No extended cycle for leading and trailing edge of DQS window (minimum window)
27:26	DQS1_GATING_WINDOW	**Controls the mask for the data strobe 1 DQS1 window leading and trailing edge.** 00: Half extended cycle for the leading and trailing edge of DQS window (maximum window) 01: Only half extended cycle for leading edge of DQS window 10: Only half extended cycle for trailing edge of DQS window 11: No extended cycle for leading and trailing edge of DQS window (minimum window)
25:13	RSV0	**Reserved**
12	PD	**Active Memory Power Down Exit Time** 0: Fast exit time (TXARD) 1: Slow exit time(TXARDS) This bit is used for DDR2 only. This bit must be 0 for DDR1.
11:9	WR	**Auto Pre-charge Write Recovery (TDAL)** These bits must be 0 for DDR1.
8	DLLRESET	**SDRAM Delay Locked Loop (DLL) Reset** 0: Normal operation 1: Normal operation with DLL reset
7	TESTMODE	**Set SDRAM to run test mode.** 0: Normal operation. 1: Test mode. The user must keep this bit at 0 if SDRAM does not support TESTMODE bit.
6:4	CAS_LATENCY	**Specifies the number of the clock cycles from the assertion of a read/write signal to the SDRAM until the first valid data on the output from the SDRAM. The valid numbers are:** 101: 1.5 for DDR1 or 5 for DDR2. 010: 2 110: 2.5 (DDR1 only) 011: 3

Bit(s)	Name	Description
		100: 4 (DDR2 only)
3	BURST_TYPE	**This register is hardwired to 0 to indicate a sequential burst type.**
2:0	BURST_LENGTH	**Indicates the burst length of the read/write transaction.** 010: 4 bursts 011: 8 bursts NOTE: 1. A burst of 4 is not allowed when user data is 64-bit while SDRAM data is 16-bit. 2. A burst of 8 is allowed in all user/SDRAM data width combination. 3. Other values for burst length are not allowed.

1000034C **DDR_CFG3** **DDR1/DDR2 controller configuration 3 register** **FFFFE412**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[18:3]															
Type	RO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[2:0]			Q_OFF	RDOS	DIS_DIFF_DQS	OCD			RTT1	ADDITIVE_LATENCY			RTT0	DS	DLL
Type	RO			RW	RW	RW	RW			RW	RW			RW	RW	RW
Reset	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	0

Bit(s)	Name	Description
31:13	RSV0	**Reserved**
12	Q_OFF	**Output Buffer Disable** 0: Enabled 1: Disabled This bit is used for DDR2 only. This bit must be 0 for DDR1.
11	RDOS	**Redundant Data Strobe (DQS)** This bit enables the redundant DQS function if supported by the SDRAM. 0: Disable 1: Enable This bit is used for DDR2 only and must be 0 for DDR1.
10	DIS_DIFF_DQS	**Disable differential DQS** 0: Enable 1: Disable This bit is used for DDR2 only and must be 0 for DDR1.
9:7	OCD	**Off-Chip Driver Impedance Calibration (OCD)** These bits support the OCD function if supported by the SDRAM. The value programmed in these register bits will be programmed into the SDRAM at EMR1 programming. Settings are vendor-dependant.
6	RTT1	**Internal Termination Resistor (RTT) bit 1** Used together with bit 2 (RTT0) to control On-Die Termination (ODT). Combine values for (RTT1, RTT0) to select ODT settings. 00: ODT disabled. 01: 75 ohm 10: 150 ohm 11: Reserved This bit is used for DDR2 only and must be 0 for DDR1.
5:3	ADDITIVE_LATENCY	**Additive Latency** 000: 0 cycle

Bit(s)	Name	Description
		001: 1 cycle 010: 2 cycles 011: 3 cycles 100: 4 cycles 101: 5 cycles Others: Reserved This bit is used for DDR2 only and must be 0 for DDR1.
2	RTT0	**Internal Termination Resistor (RTT) bit 0** Used together with bit 6 (RTT1) to control ODT. This bit is used for DDR2 only and must be 0 for DDR1.
1	DS	**SDRAM drive Strength** 0: 100% drive strength. 1: 60% drive strength.
0	DLL	**SDRAM Delay Locked Loop (DLL) Enable** 0: Disable 1: Enable

10000350 DDR_CFG4 DDR1/DDR2 controller configuration 4 register FFFFFFF4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[26:11]															
Type	RO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[10:0]											FAW				
Type	RO											RW				
Reset	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0

Bit(s)	Name	Description
31:5	RSV0	**Reserved**
4:0	FAW	**Four Activated Windows (FAW) Period** DDR2 devices impose a restriction in that no more than 4 ACTIVE commands may be issued in a given FAW period. To obtain this value, one should divide the Four Bank Activate period (TFAW) of the DDR by the clock cycle time. These bits are ignored in 4 bank devices.

10000360 DDR_DQ_DLY DDR1/DDR2 DQ delay control register 00008888

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQ_GROUP1_DELAY_SEL								DQ_GROUP0_DELAY_SEL							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQ_GROUP1_DELAY_COARSE_TUNING				DQ_GROUP1_DELAY_FINE_TUNING				DQ_GROUP0_DELAY_COARSE_TUNING				DQ_GROUP0_DELAY_FINE_TUNING			
Type	RW				RW				RW				RW			
Reset	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:24	DQ_GROUP1_DELAY_SEL	**Force Data Group 1 (MD8 to MD15) Output Delay. Valid when DQ_DLY_SEL_EN is 1.** bit7~4: for coarse-grain delay setting

Bit(s)	Name	Description
		bit3~0: for fine-grain delay setting
23:16	DQ_GROUP0_DELAY_SEL	**Force Data Group 0 (MD0 to MD7) Output Delay. Valid when DQ_DLY_SEL_EN is 1.** bit7~4: for coarse-grain delay setting bit3~0: for fine-grain delay setting
15:12	DQ_GROUP1_DELAY_COARSE_TUNING	**Data Group 1 (MD8 to MD15) Output Delay Coarse-Grain Tuning** 0x0 to 0x7: Decrease delay by 250 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 250 ps per step.
11:8	DQ_GROUP1_DELAY_FINE_TUNING	**Data Group 1 (MD8 to MD15) Output Delay Fine-Grain Tuning** 0x0 to 0x7: Decrease delay by 30 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 30 ps per step.
7:4	DQ_GROUP0_DELAY_COARSE_TUNING	**Data Group 0 (MD0 to MD7) Output Delay Coarse-Grain Tuning** 0x0 to 0x7: Decrease delay by 250 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 250 ps per step.
3:0	DQ_GROUP0_DELAY_FINE_TUNING	**Data Group 0 (MD0 to MD7) Output Delay Fine-Grain Tuning** 0x0 to 0x7: Decrease delay by 30 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 30 ps per step.

10000364 **DDR_DQS_DLY** **DDR1/DDR2 DQS delay control register** **00008888**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DQS1_DELAY_SEL								DQS0_DELAY_SEL							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DQS1_DELAY_COARSE_TUNING				DQS1_DELAY_FINE_TUNING				DQS0_DELAY_COARSE_TUNING				DQS0_DELAY_FINE_TUNING			
Type	RW				RW				RW				RW			
Reset	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:24	DQS1_DELAY_SEL	**Force Data Strobe 1 (MDQS1) Input Delay. Valid when DQS_DLY_SEL_EN is 1** bit7~4: for coarse-grain delay setting bit3~0: for fine-grain delay setting
23:16	DQS0_DELAY_SEL	**Force Data Strobe 0 (MDQS0) Input Delay. Valid when DQS_DLY_SEL_EN is 1** bit7~4: for coarse-grain delay setting bit3~0: for fine-grain delay setting
15:12	DQS1_DELAY_COARSE_TUNING	**Data Strobe 1 Input Delay Coarse-Grain Tuning** 0x0 to 0x7: Decrease delay by 250 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 250 ps per step.
11:8	DQS1_DELAY_FINE_TUNING	**Data Strobe 1 Input Delay Fine-Grain Tuning** 0x0 to 0x7: Decrease delay by 30 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 30 ps per step.
7:4	DQS0_DELAY_COARSE_TUNING	**Data Strobe 0 Input Delay Coarse-Grain Tuning** 0x0 to 0x7: Decrease delay by 250 ps per step.

Bit(s)	Name	Description
		0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 250 ps per step.
3:0	DQS0_DELAY_FINE_TUNING	**Data Strobe 0 Input Delay Fine-Grain Tuning** 0x0 to 0x7: Decrease delay by 30 ps per step. 0x8: Keep DLL delay. 0x9 to 0xF: Increase delay by 30 ps per step.

10000368 **DDR_DLL_SLV** **DDR1/DDR2 DLL slave control register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RSV0[22:7]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV0[6:0]**							**DLL_SLV_UPDATE_MODE**	**RSV1**			**DQS_DLY_SEL_EN**	**RSV2**			**DQ_DLY_SEL_EN**
Type	RO							RW	RO			RW	RO			RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:9	RSV0	**Reserved**
8	DLL_SLV_UPDATE_MODE	**Set DLL slave update mode.** 0: Update delay code only when bank is activated. 1: Continous update
7:5	RSV1	**Reserved**
4	DQS_DLY_SEL_EN	**0: DQS Input Delay decided by DLL.** 1: Force DQS Input Delay by DQS0_DELAY_SEL / DQS1_DELAY_SEL.
3:1	RSV2	**Reserved**
0	DQ_DLY_SEL_EN	**0: DQ Output Delay decided by DLL.** 1: Force DQ Output Delay by DQ_GROUP0_DELAY_SEL / DQ_GROUP1_DELAY_SEL.

1000036C **DDR_DLL_MST** **DDR1/DDR2 DLL master control register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DLL_MAS_RELOCK_EN**	**RSV0**					**DLL_MAS_BYPASS_FD**	**DLL_MAS_BYPASS_CD**	**RSV1[11:4]**							
Type	RW	RO					RW	RW	RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV1[3:0]**				**DLL_MAS_FIXED_FD**				**RSV2**		**DLL_MAS_FIXED_CD**					
Type	RO				RW				RO		RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	DLL_MAS_RELOCK_EN	**Delayed Locked Loop (DLL) Master Relock Enable** 0: Disable relocking scheme. 1: Enable relocking scheme. DLL supports restarting locking from initial value if DLL is not locked after waiting 512 cycles.
30:26	RSV0	**Reserved**
25	DLL_MAS_BYPASS_FD	**DLL Bypass Fine Grain Delay** 0: Fine-grain delay code is determined by DLL. 1: Fine-grain delay code is fixed by DLL_MAS_FIXED_FD.
24	DLL_MAS_BYPASS_CD	**DLL Bypass Coarse Grain Delay** 0: Coarse-grain delay code is determined by DLL 1: Coarse-grain delay code is fixed by DLL_MAS_FIXED_CD.
23:12	RSV1	**Reserved**
11:8	DLL_MAS_FIXED_FD	**DLL Fixed Fine Grain Delay** Specifies the fine-grain delay. The effective range is 0 to 15. Each step is about 30 ps.
7:6	RSV2	**Reserved**
5:0	DLL_MAS_FIXED_CD	**DLL Fixed Coarse Grain Delay** Specifies the coarse-grain delay. The delay = ((x-2)/4-1)*250 ps, the effective range of x is 10 to 52.

10000380 **MC_ARB_CFG** **MC 2 to 1 arbiter setting** **07FAC688**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RSV0**					**preempt_en**	**trtc_en**	**class_en**	**cls_priority[23:16]**							
Type	RO					RW	RW	RW	RW							
Reset	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**cls_priority[15:0]**															
Type	RW															
Reset	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:27	RSV0	**Reserved**
26	preempt_en	**Preemption Enable** Request preemption, higher priority requestor may change current request transaction 0: Disable Preemption 1: Enable Preemption
25	trtc_en	**Two Rate Three Color Bandwidth (TRTC) Meter Enable** 0: Disable TRTC 1: Enable TRTC
24	class_en	**QoS Classifier Enable** 0: Disable CLASS 1: Enable CLASS TRTC (0) CLASS (0) Round Robin TRTC (0) CLASS (1) Fixed Priority TRTC (1) CLASS (0) BW RR TRTC (1) CLASS (1) QoS Arb

Bit(s)	Name	Description
23:0	cls_priority	**Class Priority** This field is used for class priority for second arbitration. {BEy(3'd7), LCg(3'd6), BSy(3'd5), LSy(3'd4), BEg (3'd3), BSg (3'd2), LSg(3'd1), LCgd(3'd0)}

10000384 **MC_AG_BW** **MC Channel BW/QoS_Type/DueDate Setting** **0110FF40**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ag_wr**	**RSV0**		**ag_sel**	**RSV1**		**ag_qos_type**		**ag_duedate**							
Type	WO	RO		RW	RO		RW		RW							
Reset	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ag_pir**								**ag_cir**							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0

Bit(s)	Name	Description
31	ag_wr	**Agent Write** 0: Read 1: Write
30:29	RSV0	**Reserved**
28	ag_sel	**DMA Agent Select** Selects a DMA agent to configure. 0: CPU (Rbus0) 1: DMA (Rbus1)
27:26	RSV1	**Reserved**
25:24	ag_qos_type	**Agent QoS Type** 0: Latency critical 1: Latency sensitive (CPU) 2: Bandwidth sensitive (DMA) 3: Best Effort
23:16	ag_duedate	**Due date for latency critical agent** (unit: system bus clock cycle - system bus is 300 MHz or 225 MHz depending on bootstrap value.)
15:8	ag_pir	**Peak Information Rate for the Agent** The PIR is greater than or equal to the CIR. Bandwidth which exceeds PIR is marked red. 0x00: 0 MB/s 0x01: 8 MB/s ... 0x40: 512 MB/s ... 0xFF: 2040 MB/s (Max)
7:0	ag_cir	**Committed Information Rate for the Agent** Bandwidth which falls below the CIR is marked green. BW which exceeds the CIR but is below the EIR is marked yellow. 0x00: 0 MB/s 0x01: 8 MB/s ... 0x40: 512 MB/s (default) ... 0xFF: 2040 MB/s (Max)

10000390 **RB_DBG** **RB Debug** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[30:15]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[14:0]															rb_sel
Type	RO															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:1	RSV0	**Reserved**
0	rb_sel	**RB channel select for debug message dump**

10000394 **RB_STATE** **RB Debug State** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[20:5]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[4:0]					rb_rw	rb_state		rb_length							
Type	RO					RO	RO		RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:11	RSV0	**Reserved**
10	rb_rw	**RB channel RW**
9:8	rb_state	**RB channel State** 2'b00: IDLE 2'b01: REQ 2'b10: ACK 2'b11: DATA
7:0	rb_length	**RB channel burst length (Byte)**

10000398 **RB_BW** **RB Bandwidth** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	bw_rst	RSV0	avg_bw										peak_bw[9:6]			
Type	WO	RO	RO										RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	peak_bw[5:0]						rb_bw									
Type	RO						RO									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31	bw_rst	**Write 1 will reset BW values.**
30	RSV0	**Reserved**
29:20	avg_bw	**Average BW (MB/S)**
19:10	peak_bw	**Peak BW (MB/S)**
9:0	rb_bw	**RB channel BW (MB/S)**

1000039C **RB_LAT** **RB Latency** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	lat_rst	RSV0	avg_lat										peak_lat[9:6]			
Type	WO	RO	RO										RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	peak_lat[5:0]						rd_lat									
Type	RO						RO									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	lat_rst	**Write 1 will reset latency values**
30	RSV0	**Reserved**
29:20	avg_lat	**Average read latency (T)**
19:10	peak_lat	**Peak read latency (T)**
9:0	rd_lat	**RB channel read latency (T)**

2.6 R-Bus Controller

2.6.1 Features

– 8 channel QoS Arbiter
– Configurable Bandwidth and Duedate for each agent
– QoS classifier can be programmed for RR, BW RR, Fixed Priority and QoS Arb

2.6.2 Block Diagram

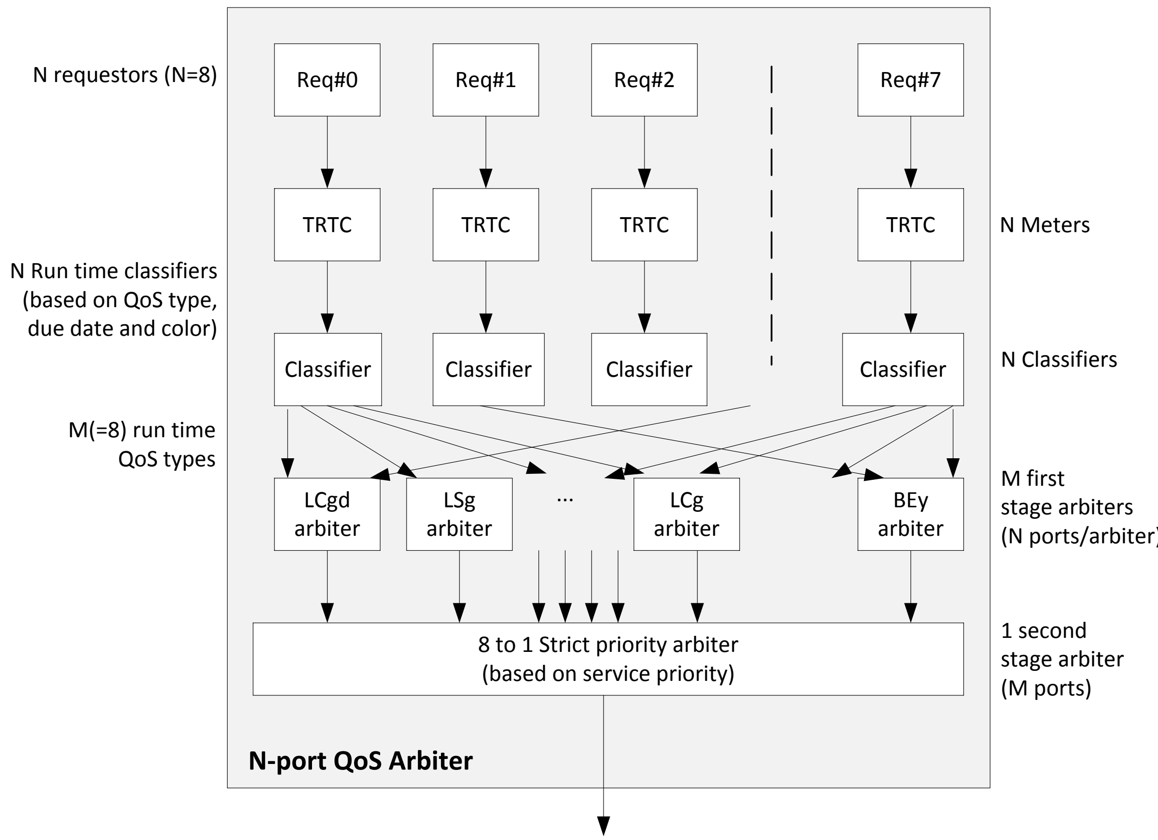


Figure 2-3 QoS Arbitration Block Diagram

2.6.3 Regsiter

Rbus_Matrix_CTRL Changes LOG

Revision	Date	Author	Change Log
0.1	2012/10/2	Lancelot	Initialization
0.2	2013/1/3	Lancelot	Add sleep count
0.2	2013/8/19	YS Xiao	Modify to as MT7621's dma_ch_csr

Module name: Rbus_Matrix_CTRL Base address: (+10000400h)

Address	Name	Widt	Register Function

		h	
10000400	DMA_ARB_CFG	32	**DMA 8 to 1 arbiter setting**
10000404	DMA_AG_BW_CFG	32	**DMA Channel BW/QoS_Type/DueDate Setting**
1000040C	DMA_ROUTE	32	**DMA Routing**
10000410	DMA_MON_AG_SEL	32	**DMA Monitor Agent Select**
10000414	DMA_STATE	32	**DMA State**
10000418	DMA_BW	32	**DMA Bandwidth**
1000041C	DMA_LAT	32	**DMA Latency**
10000420	OCP_CFG0	32	**OCP to Rbus configuration**
10000424	OCP_CFG1	32	**Read bypass write mask**
10000430	R2P_MONITOR	32	**Rbus to APbus monitor**
10000434	R2P_ERR_ADDR	32	**Rbus to APbus error address**
10000440	DYN_CFG0	32	**Dynamic cpu/ocp frequency control**
10000444	DYN_CFG1	32	**CPU sleep step frequency control**
10000448	DYN_CFG2	32	**Dyn CFG Probe**
1000044C	DYN_CFG3	32	**SI_Sleep Serial Counter Setting**
10000450	DYN_CFG4	32	**SI_Sleep Issue Count Counter**
10000454	DYN_CFG5	32	**Sleep Time Counter for SI_Sleep**
10000458	DYN_CFG6	32	**Operation Time Counter for non SI_Sleep**

10000400 DMA_ARB_CFG DMA 8 to 1 arbiter setting 04FAC688

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0					preempt_en	trtc_en	class_en	cls_priority[23:16]							
Type	RO					RW	RW	RW	RW							
Reset	0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	cls_priority[15:0]															
Type	RW															
Reset	1	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:27	RSV0	**Reserved**
26	preempt_en	**Preemption Enable** Request preemption, higher priority requestor may change current request transaction 0: Disable Preemption 1: Enable Preemption
25	trtc_en	**Two Rate Three Color Bandwidth (TRTC) Meter Enable** 0: Disable TRTC 1: Enable TRTC
24	class_en	**QoS Classifier Enable** 0: Disable CLASS 1: Enable CLASS TRTC (0) CLASS (0) Round Robin TRTC (0) CLASS (1) Fixed Priority

Bit(s)	Name	Description
		TRTC (1) CLASS (0) BW RR TRTC (1) CLASS (1) QoS Arb
23:0	cls_priority	**Class Priority** This field is used for class priority for second arbitration. {BEy(3'd7), LCg(3'd6), BSy(3'd5), LSy(3'd4), BEg (3'd3), BSg (3'd2), LSg(3'd1), LCgd(3'd0)}

10000404 **DMA_AG_BW_CFG** **DMA Channel BW/QoS_Type/DueDate Setting** **02208020**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ag_wr**	**ag_sel**			**RSV0**		**ag_qos_type**		**ag_duedate**							
Type	W1C	RW			RO		RW		RW							
Reset	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ag_pir**								**ag_cir**							
Type	RW								RW							
Reset	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit(s)	Name	Description
31	ag_wr	**Agent Write** 0: Read 1: Write
30:28	ag_sel	**DMA Agent Select** Selects a DMA agent to configure. 0: SDXC 1: GDMA 2: SPI Slave/3-Wire SPI Slave/PUTIF 3: Switch 4: WLAN 5: PCIe 6: AES 7: USB20
27:26	RSV0	**Reserved**
25:24	ag_qos_type	**Agent QoS Type** 0: Latency critical 1: Latency sensitive 2: Bandwidth sensitive (default) 3: Best Effort
23:16	ag_duedate	**Due date for latency critical agent** (unit: system bus clock cycle - system bus is 300 MHz or 225 MHz depending on bootstrap value.)
15:8	ag_pir	**Peak Information Rate for the Agent** The PIR is greater than or equal to the CIR. Bandwidth which exceeds PIR is marked red. 0x00: 0 MB/s 0x01: 4 MB/s ... 0x80: 512 MB/s (default) ... 0xFF: 1020 MB/s (Max)
7:0	ag_cir	**Committed Information Rate for the Agent** Bandwidth which falls below the CIR is marked green. BW which exceeds the CIR but is below the EIR is marked yellow.

Bit(s)	Name	Description
		0x00: 0 MB/s 0x01: 4 MB/s ... 0x20: 128 MB/s (default) ... 0xFF: 1020 MB/s (Max)

1000040C **DMA_ROUTE** **DMA Routing** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[30:15]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[14:0]															dma_route
Type	RO															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:1	RSV0	**Reserved**
0	dma_route	**DMA routing** 0: DMA will access to DRAM 1: DMA will access to CSR

10000410 **DMA_MON_AG_SEL** **DMA Monitor Agent Select** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[28:13]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[12:0]													dma_sel		
Type	RO													RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:3	RSV0	**Reserved**
2:0	dma_sel	**DMA Monitor Agent Select** Selects a DMA agent to dump DMA_STATE, DMA_BW and DMA_LAT's content. 0: SDXC 1: GDMA 2: SPI Slave/3-Wire SPI Slave/PUTIF 3: Switch 4: WLAN 5: PCIe 6: AES 7: USB20

10000414 DMA_STATE DMA State 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[20:5]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSV0[4:0]					dma_rw	dma_state		dma_length							
Type	RO					RO	RO		RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:11	RSV0	**Reserved**
10	dma_rw	**DMA channel RW state**
9:8	dma_state	**DMA channel State** 2'b00: IDLE 2'b01: REQ 2'b10: ACK 2'b11: DATA
7:0	dma_length	**DMA channel burst length (Byte) state**

10000418 DMA_BW DMA Bandwidth 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	bw_rst	RSV0	avg_bw										peak_bw[9:6]			
Type	WO	RO	RO										RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	peak_bw[5:0]						dma_bw									
Type	RO						RO									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	bw_rst	**Write 1 will reset BW values.**
30	RSV0	**Reserved**
29:20	avg_bw	**Average BW (MB/S)**
19:10	peak_bw	**Peak BW (MB/S)**
9:0	dma_bw	**DMA channel BW (MB/S)**

1000041C DMA_LAT DMA Latency 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	lat_rst	RSV0	avg_lat										peak_lat[9:6]			
Type	WO	RO	RO										RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	peak_lat[5:0]						rd_lat									
Type	RO						RO									

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	lat_rst	**Write 1 will reset latency values**
30	RSV0	**Reserved**
29:20	avg_lat	**Average read latency (T)**
19:10	peak_lat	**Peak read latency (T)**
9:0	rd_lat	**DMA channel read latency (T)**

10000420 OCP_CFG0 OCP to Rbus configuration 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[11:0]												sync_method	ocp_sync_cmd	rbus_async	rd_bypass_wr
Type	RO												RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:4	RSV0	**Reserved**
3	sync_method	**OCP Synchronization Command Method** 0: All empty (Wait until all FIFOs are empty) 1: CMD empty (Wait until the CMD FIFO is empty)
2	ocp_sync_cmd	**OCP Synchronization Command Method Enable** Remaps this RD CMD to address 0x0000_0000. Initiate DRAM control before enabling this option. 0: Disable 1: Enable
1	rbus_async	**Async Mode for RBUS** 0: Set HW to switch between sync or async mode dynamically. 1: Force RBUS to A.sync mode.
0	rd_bypass_wr	**Read Bypass Write Enable** Allows read commands to bypass write commands for OCP_IF when the address does not conflict. 0: Disable 1: Enable

10000424 OCP_CFG1 Read bypass write mask FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	rd_bypass_wr_mask[31:16]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	rd_bypass_wr_mask[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	rd_bypass_wr_mask	**Mask bit for read bypass write address**

10000430 **R2P_MONITOR** **Rbus to APbus monitor** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0															r2p_inc_clr
Type	RO															W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	r2p_err_cnt						r2p_inc_cnt									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:17	RSV0	**Reserved**
16	r2p_inc_clr	**R2APB Interrupt Clear** Write 1 to clear this interrupt.
15:10	r2p_err_cnt	**R2APB error counter**
9:0	r2p_inc_cnt	**R2APB Interrupt Countdown Timer** Sets a delay timer which begins counting down when an R2P error is detected. When the timer reaches zero the R2P interrupt is then triggered. 10'b0000000000: Disable R2P monitoring 10'b0000000001: 20 us 10'b0000000010: 40 us ... 10'b1000000000: 40 ms

10000434 **R2P_ERR_ADDR** **Rbus to APbus error address** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	r2p_err_addr[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	r2p_err_addr[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	r2p_err_addr	**R2APB address record for previous error found**

10000440 **DYN_CFG0** **Dynamic cpu/ocp frequency control** **00030A01**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RSV0													cpu_ocp_ratio		
Type	RO													RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1				cpu_fdiv				RSV2				cpu_ffrac			
Type	RO				RW				RO				RW			
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:19	RSV0	**Reserved**
18:16	cpu_ocp_ratio	**CPU OCP Ratio** The ratio between the system bus frequency and the CPU frequency. 3'b011: SYS/CPU = 1/3 3'b100: SYS/CPU = 1/4 (Not used in MT7628)
15:12	RSV1	**Reserved**
11:8	cpu_fdiv	**CPU Frequency Divider** The frequency divider is used to generate the CPU frequency. Valid values range from 1 to 15. NOTE1: CPU_FDIV must be equaled to N*CPU_FFRAC(N is a integer number) when rbus_async equal to 1'b0. NOTE2: CPU_FDIV must be larger than or equal to CPU_FFRAC when rbus_async equal to 1'b1.
7:4	RSV2	**Reserved**
3:0	cpu_ffrac	**CPU Frequency Fractional** A parameter used in conjunction with the CPU frequency divider to determine the CPU frequency. Input a value in the following equation to determine the CPU frequency. CPU frequency = PLL_FREQ*(CPU_FFRAC/CPU_FDIV) NOTE: If the chip runs in USB OHCI mode, the OCP frequency cannot be lower than 30 MHz. It means that PLL_FREQ*(CPU_FFRAC/CPU_FDIV)/CPU_OCP_RATIO >= 30 MHz.

10000444 DYN_CFG1 CPU sleep step frequency control 00230A06

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	slp_en	step_en	RSV0		step_cnt								RSV1	step_ocp_ratio		
Type	RW	RW	RO		RW								RO	RO		
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV2				step_fdiv				RSV3				step_ffrac			
Type	RO				RO				RO				RW			
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0

Bit(s)	Name	Description
31	slp_en	**Sleep Mode Enable** Enables sleep mode when MIPS SI_Sleep is asserted. 0: Disable 1: Enable Sleep Mode CPU Frequency = PLL_FREQ*(1/CPU_FDIV)
30	step_en	**Step Jump Enable** Enables step jump after MIPS exits sleep mode. The CPU will jump to the normal frequency in increments defined by STEP_FFRAC.bit[4:0] of this register. 0: Disable

Bit(s)	Name	Description
		1: Enable
29:28	RSV0	**Reserved**
27:20	step_cnt	**Step Counter** Sets the period of each step jump. When the counter counts down to zero, the CPU clock automatically changes to the next step frequency. The count period unit is 1 us.
19	RSV1	**Reserved**
18:16	step_ocp_ratio	**Step OCP Ratio (Fix to cpu_ocp_ratio)** The ratio between the system bus frequency and the CPU frequency. 3'b011: SYS/CPU = 1/3 3'b100: SYS/CPU = 1/4 (Not used in MT7628)
15:12	RSV2	**Reserved**
11:8	step_fdiv	**Step Frequency Divider (Fix to CPU_FDIV)** The frequency divider is used to generate the CPU frequency after the CPU exits from sleep mode and returns to normal operation. Valid values range from 1 to 15.
7:4	RSV3	**Reserved**
3:0	step_ffrac	**Step Frequency Fraction** The fractional size of the increment in CPU frequency after the CPU exits from sleep mode and returns to normal operation. This step is only valid when SLP_STEP_EN is enabled. FRAC_VALUE = PREVIOUS_FRAC_VALUE + STEP_FFRAC CPU Frequency = (FRAC_VALUE/CPU_FDIV)*PLL_FREQ

10000448 **DYN_CFG2** **Dyn CFG Probe** **00030A0**
1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0					dfc_fsm			RSV1			same_freq	RSV2	cpu_ocp_ratio		
Type	RO					RO			RO			RO	RO	RO		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSV3				cpu_fdiv				RSV4				cpu_ffrac			
Type	RO				RO				RO				RO			
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:27	RSV0	**Dynamic frequency controller's main FSM current state**
26:24	dfc_fsm	**Dynamic frequency controller's main FSM current state**
23:21	RSV1	**Reserved**
20	same_freq	**Indicates that the SYS and DRAM clocks are on the same frequency.**
19	RSV2	**Reserved**
18:16	cpu_ocp_ratio	**OCP ratio after changed frequency**
15:12	RSV3	**Reserved**
11:8	cpu_fdiv	**CPU fdiv after changed frequency**
7:4	RSV4	**Reserved**
3:0	cpu_ffrac	**CPU ffrac after changed frequency**

1000044C DYN_CFG3 SI_Sleep Serial Counter Setting 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	si_slp_cnt_en	RSV0			si_slp_time_unit[27:16]											
Type	RW	RO			RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	si_slp_time_unit[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	si_slp_cnt_en	**SI_Sleep Serial Counter Enable**
30:28	RSV0	**Reserved**
27:0	si_slp_time_unit	**SI_Sleep Time Counter unit** 28'h0000000: count per 1us 28'h0000001: count per 2us 28'h0000002: count per 3us ... 28'hfffffff: count per 268435456us

10000450 DYN_CFG4 SI_Sleep Issue Count Counter 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	si_slp_cnt[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	si_slp_cnt[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	si_slp_cnt	**SI_Sleep Issue Count Counter** Write to this register will clear the counter value.

10000454 DYN_CFG5 Sleep Time Counter for SI_Sleep 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	si_slp_time_unit_cnt[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	si_slp_time_unit_cnt[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	si_slp_time_unit_cnt	**Sleep Time Counter for SI_Sleep** Finally, CPU in SI_Sleep time is "si_slp_time_unit_cnt*si_slp_time_unit(us)".

Bit(s)	Name	Description
		Write to this register will clear the counter value.

10000458 **DYN_CFG6** **Operation Time Counter for non SI_Sleep** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	si_opt_time_unit_cnt[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	si_opt_time_unit_cnt[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	si_opt_time_unit_cnt	**Operation Time Counter for non SI_Sleep** Finally, CPU in non SI_Sleep time is "si_opt_time_unit_cnt*si_slp_time_unit(us)". Write to this register will clear the counter value.

2.7 MIPS CNT

2.7.1 Registers

MIPS_CNT Changes LOG

Revision	Date	Author	Change Log
0.1	2013/1/14	YuShu Xiao	Initialization

Module name: MIPS_CNT Base address: (+10000500h)

Address	Name	Width	Register Function
10000500	STCK_CNT_CFG	32	**MIPS Configuration**
10000504	CMP_CNT	32	**MIPS Compare** Sets the cutoff point for the free run counter (MIPS counter). If the free run counter equals the compare counter, then the timer circuit generates an interrupt. The interrupt remains active until the compare counter is written again.
10000508	CNT	32	**MIPS Counter** The MIPS counter (free run counter) increases by 1 every 20 us (50 KHz). The counter continues to count until it reaches the value loaded into CMP_CNT.

10000500 STCK_CNT_CFG MIPS Configuration 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[29:14]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[13:0]														EXT_STK_EN	CNT_EN
Type	RO														RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:2	RESV	
1	EXT_STK_EN	**External System Tick Enable - Selects the system tick source.** 0: Use the MIPS internal timer interrupts. 1: Use the external timer interrupt from an external MIPS counter.
0	CNT_EN	**Counter Enable - Enable the free run counter (MIPS counter).** 0: Disable 1: Enable

10000504 CMP_CNT MIPS Compare 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															

Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CMP_CNT**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	
15:0	CMP_CNT	**Compare Count**

10000508 **CNT** **MIPS Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	
15:0	CNT	**MIPS Counter**

2.8 General Purpose IO

2.8.1 Features

- Parameterized numbers of independent inputs, outputs, and inouts
- Independent polarity controls for each pin
- Independently masked edge detect interrupt on any input transition

2.8.2 Block Diagram

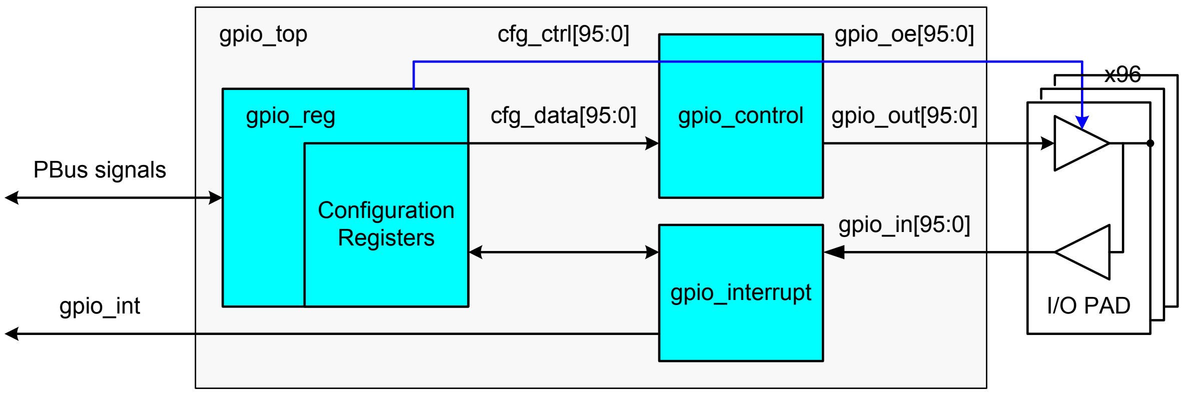


Figure 2-4 Programmable I/O Block Diagram

2.8.3 GPIO pin mapping

PAD Name	Function 0	Function 1	Function 2	Function 3	strap	pmux_group	GPIO
PAD_I2S_SDI	i2ssdi (I)	gpio (I/O)	pcmdrx (I)	antsel[5] (O)		i2s_gpio_psel[2:0]	0
PAD_I2S_SDO	i2ssdo (O)	gpio (I/O)	pcmdtx (O)	antsel[4] (O)	0	i2s_gpio_psel[2:0]	1
PAD_I2S_WS	i2sws(I/O)	gpio (I/O)	pcmclk (I/O)	antsel[3] (O)		i2s_gpio_psel[2:0]	2
PAD_I2S_CLK	i2sclk (I/O)	gpio (I/O)	pcmfs (I/O)	antsel[2] (O)		i2s_gpio_psel[2:0]	3
PAD_I2C_SCLK	i2c_sclk (I/O)	gpio (I/O)	sutif_txd (O)	ext_bgclk (I)		i2c_gpio_psel[2:0]	4
PAD_I2C_SD	i2c_sd (I/O)	gpio (I/O)	sutif_rxd (I)			i2c_gpio_psel[2:0]	5
PAD_SPI_CS1	spi_cs1 (O)	gpio (I/O)	co_clko (O)		1	spi_cs1_psel[2:0]	6
PAD_SPI_CLK	spi_clk (O)	gpio (I/O)			2	spi_gpio_psel[1:0]	7
PAD_SPI_MOSI	spi_mosi (I/O)	gpio (I/O)			3	spi_gpio_psel[1:0]	8
PAD_SPI_MISO	spi_miso (I/O)	gpio (I/O)				spi_gpio_psel[1:0]	9
PAD_SPI_CS0	spi_cs0 (O)	gpio (I/O)				spi_gpio_psel[1:0]	10
PAD_GPIO0	gpio (I/O)	gpio (I/O)	co_clko (O)	perst_n (O)	4	gpio_psel[2:0]	11
PAD_TXD0	txd0 (O)	gpio (I/O)			5	uart0_gpio_psel[2:0]	12
PAD_RXD0	rxd0 (I)	gpio (I/O)				uart0_gpio_psel[2:0]	13
PAD_MDI_TP_P1	spis_cs (I)	gpio (I/O)	w_utif[0] (I/O)	pwm_ch0 (O)		spis_gpio_psel[2:0]	14
PAD_MDI_TN_P1	spis_clk (I)	gpio (I/O)	w_utif[1] (I/O)	pwm_ch1 (O)		spis_gpio_psel[2:0]	15
PAD_MDI_RP_P1	spis_miso (O)	gpio (I/O)	w_utif[2] (I/O)	txd2 (O)		spis_gpio_psel[2:0]	16
PAD_MDI_RN_P1	spis_mosi (I)	gpio (I/O)	w_utif[3] (I/O)	rxd2 (I)		spis_gpio_psel[2:0]	17
PAD_MDI_RP_P2	pwm_ch0 (O)	gpio (I/O)	w_utif[4] (I/O)	sd_d7 (I/O)		pwm0_gpio_psel[2:0]	18
PAD_MDI_RN_P2	pwm_ch1 (O)	gpio (I/O)	w_utif[5] (I/O)	sd_d6 (I/O)		pwm1_gpio_psel[2:0]	19
PAD_MDI_TP_P2	txd2 (O)	gpio (I/O)	pwm_ch2 (O)	sd_d5 (I/O)		uart2_gpio_psel[2:0]	20
PAD_MDI_TN_P2	rxd2 (I)	gpio (I/O)	pwm_ch3 (O)	sd_d4 (I/O)		uart2_gpio_psel[2:0]	21
PAD_MDI_TP_P3	sd_wp (I)	gpio (I/O)	w_utif[10] (I/O)	w_dbgin (I)		sd_gpio_psel[2:0]	22
PAD_MDI_TN_P3	sd_cd (I)	gpio (I/O)	w_utif[11] (I/O)	w_dbgack (O)		sd_gpio_psel[2:0]	23
PAD_MDI_RP_P3	sd_d1 (I/O)	gpio (I/O)	w_utif[12] (I/O)	w_jtclk (I)		sd_gpio_psel[2:0]	24
PAD_MDI_RN_P3	sd_d0 (I/O)	gpio (I/O)	w_utif[13] (I/O)	w_jtdi (I)		sd_gpio_psel[2:0]	25
PAD_MDI_RP_P4	sd_clk (I/O)	gpio (I/O)	w_utif[14] (I/O)	w_jtdo (O)		sd_gpio_psel[2:0]	26
PAD_MDI_RN_P4	sd_cmd (I/O)	gpio (I/O)	w_utif[15] (I/O)	dbg_uart_txd (O)		sd_gpio_psel[2:0]	27
PAD_MDI_TP_P4	sd_d3 (I/O)	gpio (I/O)	w_utif[16] (I/O)	w_jtms (I)		sd_gpio_psel[2:0]	28
PAD_MDI_TN_P4	sd_d2 (I/O)	gpio (I/O)	w_utif[17] (I/O)	w_jtrst_n (I)		sd_gpio_psel[2:0]	29
PAD_EPHY_LED4_K	ephy_led4_k (O)	gpio (I/O)	w_utif_k[6] (I/O)	jtrstn_k (I)		p4_led_kn_psel[2:0]	30
PAD_EPHY_LED3_K	ephy_led3_k (O)	gpio (I/O)	w_utif_k[7] (I/O)	jtclk_k (I)		p3_led_kn_psel[2:0]	31
PAD_EPHY_LED2_K	ephy_led2_k (O)	gpio (I/O)	w_utif_k[8] (I/O)	jtms_k (I)		p2_led_kn_psel[2:0]	32
PAD_EPHY_LED1_K	ephy_led1_k (O)	gpio (I/O)	w_utif_k[9] (I/O)	jtdi_k (I)		p1_led_kn_psel[2:0]	33
PAD_EPHY_LED0_K	ephy_led0_k (O)	gpio (I/O)		jtdo_k (I/O)		p0_led_kn_psel[2:0]	34
PAD_WLED_K	wled_k (I/O)	gpio (I/O)				wled_kn_psel[2:0]	35
PAD_PERST_N	perst_n (O)	gpio (I/O)			6	prest_gpio_psel[1:0]	36
PAD_CO_CLKO	co_clko (O)	gpio (I/O)			7	rclk_gpio_psel[1:0]	37
PAD_WDT_RST_N	wdt (I/O)	gpio (I/O)				wdt_gpio_psel[1:0]	38
PAD_EPHY_LED4_N	ephy_led4_n (O)	gpio (I/O)	w_utif_n[6] (I/O)	jtrstn_n (I)		p4_led_gpio_psel[2:0]	39
PAD_EPHY_LED3_N	ephy_led3_n (O)	gpio (I/O)	w_utif_n[7] (I/O)	jtclk_n (I)		p3_led_gpio_psel[2:0]	40
PAD_EPHY_LED2_N	ephy_led2_n (O)	gpio (I/O)	w_utif_n[8] (I/O)	jtms_n (I)		p2_led_gpio_psel[2:0]	41
PAD_EPHY_LED1_N	ephy_led1_n (O)	gpio (I/O)	w_utif_n[9] (I/O)	jtdi_n (I)		p1_led_gpio_psel[2:0]	42
PAD_EPHY_LED0_N	ephy_led0_n (O)	gpio (I/O)		jtdo_n (I/O)		p0_led_gpio_psel[2:0]	43
PAD_WLED_N	wled_n (I/O)	gpio (I/O)				wled_gpio_psel[2:0]	44
PAD_TXD1	txd1 (O)	gpio (I/O)	pwm_ch0 (O)	antsel[1] (O)	8	uart1_gpio_psel[2:0]	45
PAD_RXD1	rxd1 (I)	gpio (I/O)	pwm_ch1 (O)	antsel[0] (O)		uart1_gpio_psel[2:0]	46

2.8.4 Register

GPIO Changes LOG

Revision	Date	Author	Change Log
0.1	2012/6/21	YuShu Xiao	Initialization

Module name: GPIO Base address: (+10000600h)

Address	Name	Width	Register Function
10000600	GPIO_CTRL_0	32	**GPIO0 to GPIO31 direction control register** These direction control registers are used to select the data direction of the GPIO pin. The value driven onto the GPIO pins, are controlled by the GPIO_POL_x, and GPIO_DATA_x registers.
10000604	GPIO_CTRL_1	32	**GPIO32 to GPIO63 direction control register** These direction control registers are used to select the data direction of the GPIO pin. The value driven onto the GPIO pins, are controlled by the GPIO_POL_x, and GPIO_DATA_x registers.
10000608	GPIO_CTRL_2	32	**GPIO64 to GPIO95 direction control register** These direction control registers are used to select the data direction of the GPIO pin. The value driven onto the GPIO pins, are controlled by the GPIO_POL_x, and GPIO_DATA_x registers.
10000610	GPIO_POL_0	32	**GPIO0 to GPIO31 polarity control register** These polarity control registers are used to control the polarity of the data is driven on or read from the GPIO pin.
10000614	GPIO_POL_1	32	**GPIO32 to GPIO63 polarity control register** These polarity control registers are used to control the polarity of the data is driven on or read from the GPIO pin.
10000618	GPIO_POL_2	32	**GPIO64 to GPIO95 polarity control register** These polarity control registers are used to control the polarity of the data is driven on or read from the GPIO pin.
10000620	GPIO_DATA_0	32	**GPIO0 to GPIO31 data register** These data registers store current GPIO data value for GPIO input mode, or output driven value for GPIO output mode. Bit position stand for correspondent GPIO pin.
10000624	GPIO_DATA_1	32	**GPIO32 to GPIO63 data register** These data registers store current GPIO data value for GPIO input mode, or output driven value for GPIO output mode. Bit position stand for correspondent GPIO pin.
10000628	GPIO_DATA_2	32	**GPIO64 to GPIO95 data register** These data registers store current GPIO data value for GPIO input mode, or output driven value for GPIO output mode. Bit position stand for correspondent GPIO pin.
10000630	GPIO_DSET_0	32	**GPIO0 to GPIO31 data set register** These data set registers are used to set bits in the GPIO_DATA_x registers.
10000634	GPIO_DSET_1	32	**GPIO32 to GPIO63 data set register** These data set registers are used to set bits in the GPIO_DATA_x registers.
10000638	GPIO_DSET_2	32	**GPIO64 to GPIO95 data set register** These data set registers are used to set bits in the GPIO_DATA_x registers.
10000640	GPIO_DCLR_0	32	**GPIO0 to GPIO31 data clear register** These data set registers are used to clear bits in the GPIO_DATA_x registers.
10000644	GPIO_DCLR_1	32	**GPIO32 to GPIO63 data clear register**

			These data set registers are used to clear bits in the GPIO_DATA_x registers.
10000648	**GPIO_DCLR_2**	32	**GPIO64 to GPIO95 data clear register** These data set registers are used to clear bits in the GPIO_DATA_x registers.
10000650	**GINT_REDGE_0**	32	**GPIO0 to GPIO31 rising edge interrupt enable register** These registers are used to enable the condition of rising edge triggered interrupt.
10000654	**GINT_REDGE_1**	32	**GPIO32 to GPIO63 rising edge interrupt enable register** These registers are used to enable the condition of rising edge triggered interrupt.
10000658	**GINT_REDGE_2**	32	**GPIO64 to GPIO95 rising edge interrupt enable register** These registers are used to enable the condition of rising edge triggered interrupt.
10000660	**GINT_FEDGE_0**	32	**GPIO0 to GPIO31 falling edge interrupt enable register** These registers are used to enable the condition of falling edge triggered interrupt.
10000664	**GINT_FEDGE_1**	32	**GPIO32 to GPIO63 falling edge interrupt enable register** These registers are used to enable the condition for falling edge triggered interrupt.
10000668	**GINT_FEDGE_2**	32	**GPIO64 to GPIO95 falling edge interrupt enable register** These registers are used to enable the condition of falling edge triggered interrupt.
10000670	**GINT_HLVL_0**	32	**GPIO0 to GPIO31 high level interrupt enable register** These registers are used to enable the condition of high level triggered interrupt. The bit in this register and the corresponded bit in GINT_LLVL_0 can not be set to 1 at the same time.
10000674	**GINT_HLVL_1**	32	**GPIO32 to GPIO63 high level interrupt enable register** These registers are used to enable the condition of high level triggered interrupt. The bit in this register and the corresponded bit in GINT_LLVL_1 can not be set to 1 at the same time.
10000678	**GINT_HLVL_2**	32	**GPIO64 to GPIO95 high level interrupt enable register** These registers are used to enable the condition of high level triggered interrupt. The bit in this register and the corresponded bit in GINT_LLVL_2 can not be set to 1 at the same time.
10000680	**GINT_LLVL_0**	32	**GPIO0 to GPIO31 low level interrupt enable register** These registers are used to enable the condition of low level triggered interrupt. The bit in this register and the corresponded bit in GINT_HLVL_0 can not be set to 1 at the same time.
10000684	**GINT_LLVL_1**	32	**GPIO32 to GPIO63 low level interrupt enable register** These registers are used to enable the condition of low level triggered interrupt. The bit in this register and the corresponded bit in GINT_HLVL_1 can not be set to 1 at the same time.
10000688	**GINT_LLVL_2**	32	**GPIO64 to GPIO95 low level interrupt enable register** These registers are used to enable the condition of low level triggered interrupt. The bit in this register and the corresponded bit in GINT_HLVL_2 can not be set to 1 at the same time.
10000690	**GINT_STAT_0**	32	**GPIO0 to GPIO31 interrupt status register** These registers are used to record the GPIO current interrupt status.
10000694	**GINT_STAT_1**	32	**GPIO32 to GPIO63 interrupt status register** These registers are used to record the GPIO current interrupt status.
10000698	**GINT_STAT_2**	32	**GPIO64 to GPIO95 interrupt status register**

			These registers are used to record the GPIO current interrupt status.
100006A0	GINT_EDGE_0	32	**GPIO0 to GPIO31 edge status register** These registers are used to record the GPIO current interrupt's edge status. These registers are useful only in edge triggered interrupt.
100006A4	GINT_EDGE_1	32	**GPIO32 to GPIO63 edge status register** These registers are used to record the GPIO current interrupt's edge status. These registers are useful only in edge triggered interrupt.
100006A8	GINT_EDGE_2	32	**GPIO64 to GPIO95 edge status register** These registers are used to record the GPIO current interrupt's edge status. These registers are useful only in edge triggered interrupt.

10000600 GPIO_CTRL_0 GPIO0 to GPIO31 direction control register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIOCTRL0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIOCTRL0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIOCTRL0	**GPIO Pin Direction** 0: GPIO input mode 1: GPIO output mode

10000604 GPIO_CTRL_1 GPIO32 to GPIO63 direction control register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIOCTRL1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIOCTRL1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIOCTRL1	**GPIO Pin Direction** 0: GPIO input mode 1: GPIO output mode

10000608 GPIO_CTRL_2 GPIO64 to GPIO95 direction control register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIOCTRL2[31:16]															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIOCTRL2[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIOCTRL2	**GPIO Pin Direction** 0: GPIO input mode 1: GPIO output mode

10000610 **GPIO_POL_0** **GPIO0 to GPIO31 polarity control register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**GPIOPOL0[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIOPOL0[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIOPOL0	**GPIO Data Polarity** 0: Data is non-inverted 1: Data is inverted

10000614 **GPIO_POL_1** **GPIO32 to GPIO63 polarity control register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**GPIOPOL1[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIOPOL1[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIOPOL1	**GPIO Data Polarity** 0: Data is non-inverted 1: Data is inverted

10000618 **GPIO_POL_2** **GPIO64 to GPIO95 polarity control register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**GPIOPOL2[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIOPOL2[15:0]**															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIOPOL2	**GPIO Data Polarity** 0: Data is non-inverted 1: Data is inverted

10000620	GPIO_DATA_0	GPIO0 to GPIO31 data register	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODATA0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODATA0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIODATA0	**GPIO Data**

10000624	GPIO_DATA_1	GPIO32 to GPIO63 data register	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODATA1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODATA1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIODATA1	**GPIO Data**

10000628	GPIO_DATA_2	GPIO64 to GPIO95 data register	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODATA2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODATA2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GPIODATA2	**GPIO Data**

10000630 GPIO_DSET_0 GPIO0 to GPIO31 data set register FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODSET0[31:16]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODSET0[15:0]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	GPIODSET0	**GPIO Data Set** 1: Set the GPIO_DATA_0 register 0: No effect

10000634 GPIO_DSET_1 GPIO32 to GPIO63 data set register FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODSET1[31:16]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODSET1[15:0]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	GPIODSET1	**GPIO Data Set** 1: Set the GPIO_DATA_1 register 0: No effect

10000638 GPIO_DSET_2 GPIO64 to GPIO95 data set register FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODSET2[31:16]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODSET2[15:0]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	GPIODSET2	**GPIO Data Set** 1: Set the GPIO_DATA_2 register 0: No effect

10000640 GPIO_DCLR_0 GPIO0 to GPIO31 data clear register FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODCLR0[31:16]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODCLR0[15:0]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	GPIODCLR0	**GPIO Data Clear** 1: Clear the GPIO_DATA_0 register 0: No effect

10000644	GPIO_DCLR_1	GPIO32 to GPIO63 data clear register	FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODCLR1[31:16]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODCLR1[15:0]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	GPIODCLR1	**GPIO Data Clear** 1: Clear the GPIO_DATA_1 register 0: No effect

10000648	GPIO_DCLR_2	GPIO64 to GPIO95 data clear register	FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GPIODCLR2[31:16]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIODCLR2[15:0]															
Type	WO															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:0	GPIODCLR2	**GPIO Data Clear** 1: Clear the GPIO_DATA_2 register 0: No effect

10000650	GINT_REDGE_0	GPIO0 to GPIO31 rising edge interrupt enable register	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTREDGE0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTREDGE0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTREDGE0	**GPIO Rising Edge Interrupt Enable** 1: Enable rising edge triggered 0: Disable rising edge triggered

10000654 GINT_REDGE_1 GPIO32 to GPIO63 rising edge interrupt enable register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTREDGE1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTREDGE1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTREDGE1	**GPIO Rising Edge Interrupt Enable** 1: Enable rising edge triggered 0: Disable rising edge triggered

10000658 GINT_REDGE_2 GPIO64 to GPIO95 rising edge interrupt enable register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTREDGE2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTREDGE2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTREDGE2	**GPIO Rising Edge Interrupt Enable** 1: Enable rising edge triggered 0: Disable rising edge triggered

10000660 GINT_FEDGE_0 GPIO0 to GPIO31 falling edge interrupt enable register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTFEDGE0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTFEDGE0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTFEDGE0	**GPIO Falling Edge Interrupt Enable** 1: Enable falling edge triggered 0: Disable falling edge triggered

10000664 **GINT_FEDGE_1** **GPIO32 to GPIO63 falling edge interrupt enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTFEDGE1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTFEDGE1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTFEDGE1	**GPIO Falling Edge Interrupt Enable** 1: Enable falling edge triggered 0: Disable falling edge triggered

10000668 **GINT_FEDGE_2** **GPIO64 to GPIO95 falling edge interrupt enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTFEDGE2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTFEDGE2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTFEDGE2	**GPIO Falling Edge Interrupt Enable** 1: Enable falling edge triggered 0: Disable falling edge triggered

10000670 **GINT_HLVL_0** **GPIO0 to GPIO31 high level interrupt enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTHLVL0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTHLVL0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTHLVL0	**GPIO High Level Interrupt Enable**

Bit(s)	Name	Description
		1: Enable high level triggered 0: Disable high level triggered

10000674 GINT_HLVL_1 GPIO32 to GPIO63 high level interrupt enable register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTHLVL1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTHLVL1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTHLVL1	**GPIO High Level Interrupt Enable** 1: Enable high level triggered 0: Disable high level triggered

10000678 GINT_HLVL_2 GPIO64 to GPIO95 high level interrupt enable register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTHLVL2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTHLVL2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTHLVL2	**GPIO High Level Interrupt Enable** 1: Enable high level triggered 0: Disable high level triggered

10000680 GINT_LLVL_0 GPIO0 to GPIO31 low level interrupt enable register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTLLVL0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTLLVL0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTLLVL0	**GPIO Low Level Interrupt Enable** 1: Enable low level triggered 0: Disable low level triggered

10000684 **GINT_LLVL_1** **GPIO32 to GPIO63 low level interrupt enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTLLVL1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTLLVL1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTLLVL1	**GPIO Low Level Interrupt Enable** 1: Enable low level triggered 0: Disable low level triggered

10000688 **GINT_LLVL_2** **GPIO64 to GPIO95 low level interrupt enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTLLVL2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTLLVL2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTLLVL2	**GPIO Low Level Interrupt Enable** 1: Enable low level triggered 0: Disable low level triggered

10000690 **GINT_STAT_0** **GPIO0 to GPIO31 interrupt status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTSTAT0[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTSTAT0[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTSTAT0	**GPIO Interrupt Status** 1: Interrupt is detected 0: Interrupt is not detected

10000694 **GINT_STAT_1** **GPIO32 to GPIO63 interrupt status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTSTAT1[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTSTAT1[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTSTAT1	**GPIO Interrupt Status** 1: Interrupt is detected 0: Interrupt is not detected

10000698 **GINT_STAT_2** **GPIO64 to GPIO95 interrupt status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTSTAT2[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTSTAT2[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTSTAT2	**GPIO Interrupt Status** 1: Interrupt is detected 0: Interrupt is not detected

100006A0 **GINT_EDGE_0** **GPIO0 to GPIO31 edge status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTEDGE0[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTEDGE0[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTEDGE0	**GPIO Interrupt Edge Status** 1: Rising edge 0: Falling edge

100006A4 **GINT_EDGE_1** **GPIO32 to GPIO63 edge status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTEDGE1[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTEDGE1[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTEDGE1	**GPIO Interrupt Edge Status** 1: Rising edge 0: Falling edge

100006A8	GINT_EDGE_2	GPIO64 to GPIO95 edge status register	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GINTEDGE2[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GINTEDGE2[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	GINTEDGE2	**GPIO Interrupt Edge Status** 1: Rising edge 0: Falling edge

2.9 SPI Slave

2.9.1 SPI Slave Control

spis_intf Changes LOG

Revision	Date	Author	Change Log
0.1	2013/9/23	Kaiping Yen	Initialization

Module name: spis_intf Base address: (+0h)

Address	Name	Width	Register Function
00000000	REG00	32	SPI Slave Register 00
00000004	REG01	32	SPI Slave Register 01
00000008	REG02	32	SPI Slave Register 02
0000000C	REG03	32	SPI Slave Register 03
00000010	REG04	32	SPI Slave Register 04

00000000 REG00 SPI Slave Register 00 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	bus_read_data[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	bus_read_data[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	bus_read_data	SPI Slave Register 00 for bus read data

00000004 REG01 SPI Slave Register 01 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	bus_write_data[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	bus_write_data[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	bus_write_data	SPI Slave Register 01 for bus write data

00000008 REG02 SPI Slave Register 02 0000000

0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	bus_address[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	bus_address[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	bus_address	**SPI Slave Register 02 for bus address** This address must be physical address

0000000C **REG03** **SPI Slave Register 03** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	reg03_31_5[26:11]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reg03_31_5[10:0]											bus_pb_rb_sel	reg03_3	bus_size		bus_r_w
Type	RW											RW	RW	RW		RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:5	reg03_31_5	**reg03[31:5] reserved bit**
4	bus_pb_rb_sel	**Bus interface selection** 0: Bus transaction is asserted by Rbus master interface, can access DRAM and peripheral registers 1: Bus transaction is asserted by Pbus master interface, can peripheral registers only
3	reg03_3	**reg03[3] reserved bit**
2:1	bus_size	**Bus access size** 00: reserved 01: reserved 10: word (4bytes) 11: reserved
0	bus_r_w	**Bus access type** 0: read 1: write

00000010 **REG04** **SPI Slave Register 04** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																bus_bu

																sy
Type																RO
Reset																0

Bit(s)	Name	Description
0	bus_busy	**Bus (Internal Rbus/Pbus Master) interface status** 0: SPIS bus interface is idle for next access command 1: SPIS bus interface is busy

2.9.2 Regsiters

spis_pbslv Changes LOG

Revision	Date	Author	Change Log
0.1	2013/9/23	Kaiping Yen	Initialization

Module name: spis_pbslv Base address: (+10000700h)

Address	Name	Width	Register Function
10000700	SPIS_REG0	32	**SPI Slave Register 0**
10000704	SPIS_REG1	32	**SPI Slave Register 1**
10000708	SPIS_REG2	32	**SPI Slave Register 2**
1000070C	SPIS_REG3	32	**SPI Slave Register 3**
10000710	SPIS_REG4	32	**SPI Slave Register 4**
10000740	SPIS_CFG	32	**SPI Slave Configuration**

10000700 SPIS_REG0 SPI Slave Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	spis_reg0[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	spis_reg0[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	spis_reg0	**SPI Slave Register 0**

10000704 SPIS_REG1 SPI Slave Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	spis_reg1[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	spis_reg1[15:0]															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	spis_reg1	**SPI Slave Register 1**

10000708 SPIS_REG2 SPI Slave Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	spis_reg2[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	spis_reg2[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	spis_reg2	**SPI Slave Register 2**

1000070C SPIS_REG3 SPI Slave Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	spis_reg3[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	spis_reg3[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	spis_reg3	**SPI Slave Register 3**

10000710 SPIS_REG4 SPI Slave Register 4 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	spis_reg4[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	spis_reg4[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	spis_reg4	**SPI Slave Register 4**

10000740 SPIS_CFG SPI Slave Configuration 0000000

0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															spis_mode	
Type															RW	
Reset															0	0

Bit(s)	Name	Description
1:0	spis_mode	**SPI slave clock polarity and phase configuration** 2'b00: CPOL=0, CPHA=0 2'b01: CPOL=0, CPHA=1 2'b10: CPOL=1, CPHA=0 2'b11: CPOL=1, CPHA=1

2.10 I^2C Controller

2.10.1 Features

- Programmable I^2C bus clock rate
- Supports the Synchronous Inter-Integrated Circuits (I^2C) serial protocol
- Bi-directional data transfer
- Programmable address width up to 8 bits
- Sequential byte read or write capability
- Device address and data address can be transmitted for device, page and address selection
- Supports Standard mode and Fast mode

2.10.2 List of Registers

I2C Changes LOG

Revision	Date	Author	Change Log
0.1	2012/10/3	Evan Chou	Initialization

Module name: I2C Base address: (+10000900h)

Address	Name	Width	Register Function
10000908	SM0CFG0	32	SERIAL INTERFACE MASTER 0 CONFIG 0 REGISTER
10000910	SM0DOUT	32	SERIAL INTERFACE MASTER 0 DATAOUT REGISTER
10000914	SM0DIN	32	SERIAL INTERFACE MASTER 0 DATAIN REGISTER
10000918	SM0ST	32	SERIAL INTERFACE MASTER 0 STATUS REGISTER
1000091C	SM0AUTO	32	SERIAL INTERFACE MASTER 0 AUTO-MODE REGISTER
10000920	SM0CFG1	32	SERIAL INTERFACE MASTER 0 CONFIG 1 REGISTER
10000928	SM0CFG2	32	SERIAL INTERFACE MASTER 0 CONFIG 2 REGISTER
10000940	SM0CTL0	32	Serial interface master 0 control 0 register
10000944	SM0CTL1	32	Serial interface master 0 control 1 register
10000950	SM0D0	32	Serial interface master 0 data 0 register
10000954	SM0D1	32	Serial interface master 0 data 1 register
1000095C	PINTEN	32	Peripheral interrupt enable register
10000960	PINTST	32	Peripheral interrupt status register
10000964	PINTCL	32	Peripheral interrupt clear register

10000908 SM0CFG0 SERIAL INTERFACE MASTER 0 CONFIG 0 REGISTER 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										SM0_DEVADDR						
Type										RW						
Reset										0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
6:0	SM0_DEVADDR	**Device address for transmission**

10000910 **SM0DOUT** **SERIAL INTERFACE MASTER 0 DATAOUT REGISTER** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SM0_DATAOUT**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SM0_DATAOUT	**Data out register for auto mode**

10000914 **SM0DIN** **SERIAL INTERFACE MASTER 0 DATAIN REGISTER** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SM0_DATAIN**							
Type									RO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SM0_DATAIN	**Data in register for auto mode**

10000918 **SM0ST** **SERIAL INTERFACE MASTER 0 STATUS REGISTER** **00000002**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**SM0_RDATA_RDY**	**SM0_WDATA_EMPTY**	**SM0_BUSY**
Type														RW	RW	RO
Reset														0	1	0

Bit(s)	Name	Description
2	SM0_RDATA_RDY	**I2C read data is ready**
1	SM0_WDATA_EMPTY	**I2C data output register is empty**

Bit(s)	Name	Description
0	SM0_BUSY	**State machine is busy**

10000091C **SM0AUTO** **SERIAL INTERFACE MASTER 0 AUTO-MODE REGISTER** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SM0_START_RW
Type																RW
Reset																0

Bit(s)	Name	Description
0	SM0_START_RW	**Written with 1 to start a read transaction, and 0 to start a write transaction. This bit is only valid at auto mode.**

10000920 **SM0CFG1** **SERIAL INTERFACE MASTER 0 CONFIG 1 REGISTER** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											SM0_BYTECNT					
Type											RW					
Reset											0	0	0	0	0	0

Bit(s)	Name	Description
5:0	SM0_BYTECNT	**The value + 1 indicateds the number of data bytes for sequential reads/writes. (word address is included in data bytes)**

10000928 **SM0CFG2** **SERIAL INTERFACE MASTER 0 CONFIG 2 REGISTER** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SM0_IS_AUTOMODE
Type																RW
Reset																0

Bit(s)	Name	Description
0	SM0_IS_AUTOMODE	**Set 1 to configure auto mode**

10000940 **SM0CTL0** **Serial interface master 0 control 0 register** **00008000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SM0_ODRAIN**	**RESV0**			**SM0_CLK_DIV**											
Type	RW	RO			RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SIF_VSYNC**	**RESV1**	**SM0_VSYNC_MODE**		**RESV2**							**SM0_CS_STATUS**	**SM0_SCL_STATE**	**SM0_SDA_STATE**	**SM0_EN**	**SM0_SCL_STRECH**
Type	RO	RO	RW		RO							RO	RO	RO	RW	RW
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	SM0_ODRAIN	**Open-drain output configuration** 0: When SIF output is logic 1, the output is pulled high by outer devices. SIF output is open-drained. 1: When SIF output is logic 1, the output is pulled high by SIF master 0.
30:28	RESV0	
27:16	SM0_CLK_DIV	**SIF master 0 clock divide value** This is used to set the divider to generate expected SCL.
15	SIF_VSYNC	
14	RESV1	
13:12	SM0_VSYNC_MODE	**Restrict SIF master 0 trigger within VSYNC pulse** 00: Disable 01: Allow triggered in VSYNC pulse 10: Allow triggered at VSYNC rising edge
11:5	RESV2	
4	SM0_CS_STATUS	**Clock stretching status** 0: no clock stretching 1: clock stretching
3	SM0_SCL_STATE	**SCL value on the bus**
2	SM0_SDA_STATE	**SDA value on the bus**
1	SM0_EN	**SIF master 0 enable bit** 0: Disable SIF master 0. 1: Enable SIF master 0.
0	SM0_SCL_STRECH	**Clock stretching enable** 0: Not allow slaves hold SCL 1: Allow slaves hold SCL

10000944 **SM0CTL1** **Serial interface master 0 control 1 register** **0000000**

0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									SM0_ACK							
Type									RO							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						SM0_PGLEN				SM0_MODE						SM0_TRI
Type						RW				RW						RW
Reset						0	0	0		0	0	0				0

Bit(s)	Name	Description
23:16	SM0_ACK	**Acknowledge bits** ACK[7:0] is acknowledge of 8 bytes of data
10:8	SM0_PGLEN	**Page length** Page length of sequential read/write. The maximum is 8 bytes. Set 0 as 1 byte.
6:4	SM0_MODE	**SIF master mode** 001: Start 010: Write data 011: Stop 100: Read data with no ack for final byte 101: Read data with ack
0	SM0_TRI	**Trigger serial interface** 0: Read back as serial interface is idle. 1: Set 1 to trigger this serial interface. Read back as serial interface is busy.

10000950 SM0D0 Serial interface master 0 data 0 register FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SM0_DATA3								SM0_DATA2							
Type	RW								RW							
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SM0_DATA1								SM0_DATA0							
Type	RW								RW							
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit(s)	Name	Description
31:24	SM0_DATA3	**Serial interface data byte 3**
23:16	SM0_DATA2	**Serial interface data byte 2**
15:8	SM0_DATA1	**Serial interface data byte 1**
7:0	SM0_DATA0	**Serial interface data byte 0**

10000954 SM0D1 Serial interface master 0 data 1 register FFFFFFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SM0_DATA7								SM0_DATA6							
Type	RW								RW							
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SM0_DATA5								SM0_DATA4							

Type	RW								RW							
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit(s)	Name	Description
31:24	SM0_DATA7	**Serial interface data byte 7**
23:16	SM0_DATA6	**Serial interface data byte 6**
15:8	SM0_DATA5	**Serial interface data byte 5**
7:0	SM0_DATA4	**Serial interface data byte 4**

10000095C **PINTEN** **Peripheral interrupt enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SM0_INT_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	SM0_INT_EN	**Serial interface master 0 interrupt enable**

10000960 **PINTST** **Peripheral interrupt status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SM0_INT_ST
Type																WS
Reset																0

Bit(s)	Name	Description
0	SM0_INT_ST	**Serial interface master 0 interrupt status**

10000964 **PINTCL** **Peripheral interrupt clear register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SM0_I

															NT_ CL
Type															W1 C
Reset															0

Bit(s)	Name	Description
0	SM0_INT_CL	**Serial interface master 0 interrupt clear**

2.11 I2S Controller

2.11.1 Features

– I2S transmitter/receiver, which can be configured as master or slave.
– Supports 16-bit data, sampling rates of 8 kHz, 16 kHz, 22.05 kHz, 44.1 kHz, and 48 kHz
– Support stereo audio data transfer.
– 32-byte FIFO are available for data transmission.
– Supports GDMA access
– Supports 12 Mhz bit clock from external source (when in slave mode)

2.11.2 Block Diagram

The I^2S transmitter block diagram is shown as below.

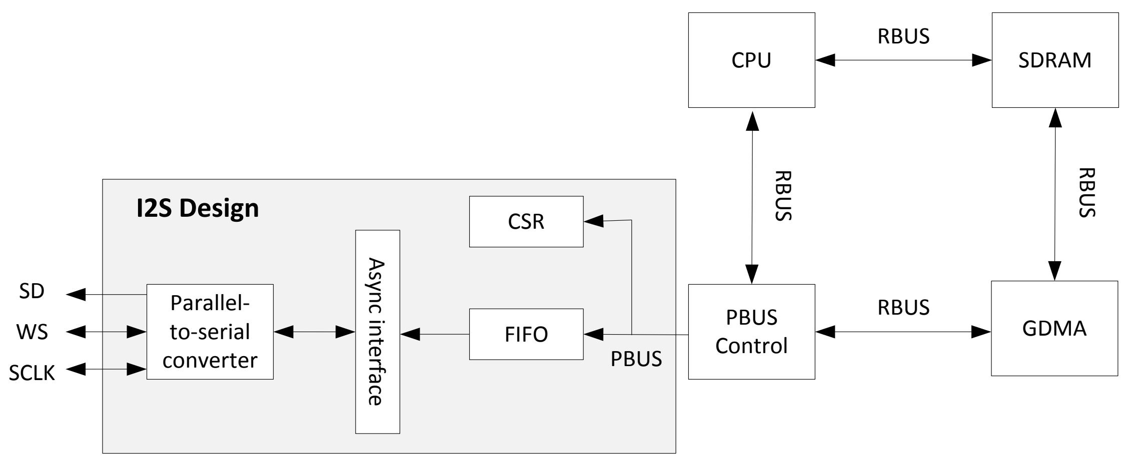


Figure 2-5 I^2S Transmitter Block Diagram

The I^2S interface consists of two separate cores, a transmitter and a receiver. Both can operate in either master or slave mode. The transmitter is only shown here in master or slave mode.

I^2S Signal Timing For I^2S Data Format

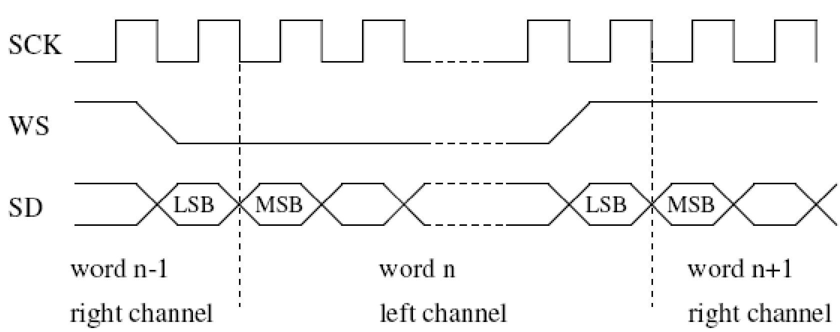


Figure 2-6 I2S Transmit/Receive

Serial data is transmitted in 2's complement with the MSB first. The transmitter always sends the MSB of the next word one clock period after the WS changes. Serial data sent by the transmitter may be synchronized with either the trailing (HIGH-to-LOW) or the leading (LOW-to-HIGH) edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions when transmitting data that is synchronized with the leading edge.

The word select line indicates the channel being transmitted:

- WS = 0; channel 1 (left)
- WS = 1; channel 2 (right)

WS may change either on a trailing or leading edge of the serial clock, but it doesn't need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The WS line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next Word.

2.11.3 Registers

I2S Changes LOG

Revision	Date	Author	Change Log
0.1	2014/1/12	Ken Wu	Initialization

Module name: I2S Base address: (+10000A00h)

Address	Name	Width	Register Function
10000A00	I2S_CFG	32	**I2S Configuration** I2S Tx/Rx Configuration Register
10000A04	INT_STATUS	32	**Interrupt Status** I2S Interrupt Status
10000A08	INT_EN	32	**Interrupt Enable** I2S Interrupt Enable Control Register
10000A0C	FF_STATUS	32	**FIFO Status** I2S Tx/Rx FIFO Status
10000A10	TX_FIFO_WREG	32	**Transmit FIFO Write to Register** Tx Write Data Buffer
10000A14	RX_FIFO_RREG	32	**Receive FIFO Read Register** DRAM PAD CONTROL 3
10000A18	I2S_CFG1	32	**I2S Configuration 1** I2S Loopback Test Control Register
10000A20	DIVCOMP_CFG	32	**Integer Part of the Dividor Register 1** Integer Part of the Dividor Register
10000A28	DIVINT_CFG	32	**Integer Part of the Dividor Register 2** Integer Part of the Dividor Register

10000A00 I2S_CFG I2S Configuration 00014040

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	I2S_EN	DMA_E	LITTIE	SYS_E				TX_EN				RX_EN		NORM	DATA_	SLAV

		N	_ENDIAN_DATA_FMT	NDIAN										_24BIT	24BIT	E_MODE
Type	RW	RW	RW	RW				RW				RW		RW	RW	RW
Reset	0	0	0	0				0				0		0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_FF_THRES								TX_FF_THRES							
Type	RW								RW							
Reset	0	1	0	0					0	1	0	0				

Bit(s)	Name	Description
31	I2S_EN	**I2S Enable** Enables I2S. When disabled, all I2S control registers are cleared to their initial values. 0: Disable 1: Enable
30	DMA_EN	**DMA Enable** Enables DMA access. 0: Disable 1: Enable
29	LITTIE_ENDIAN_DATA_FMT	**Little endian audio data** 0: big endian audio data format 1: little endian audio data format
28	SYS_ENDIAN	**System endian setting.** 0: Little endian 1: Big endian
24	TX_EN	**Transmitter on/off control** 0: Disable 1: Enable
20	RX_EN	**Receiver on/off control** 0: Disable 1: Enable
18	NORM_24BIT	**24-bit data format** 0: compact data format 1: normal data format
17	DATA_24BIT	**I2S data width** 0: 16-bit data 1: 24-bit data
16	SLAVE_MODE	**Sets master or slave mode.** 0: Master: using internal clock 1: Slave: using external clock
15:12	RX_FF_THRES	**Rx FIFO Threshold** When the threshold is reached, the host/DMA is notified to fill FIFO. 2<RX_FF_THRES<6 (unit: word)
7:4	TX_FF_THRES	**Tx FIFO Threshold** When the threshold is reached, the host/DMA is notified to fill FIFO. 2<TX_FF_THRES<6 (unit: word)

10000A04 **INT_STATUS** **Interrupt Status** **0000000**

0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RX_DMA_FAULT	RX_OVRUN	RX_UNRUN	RX_THRES	TX_DMA_FAULT	TX_OVRUN	TX_UNRUN	TX_THRES
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	RX_DMA_FAULT	**Rx DMA Fault Detected Interrupt** Asserts when a fault is detected in Rx DMA signals.
6	RX_OVRUN	**Rx Overrun Interrupt** Asserts when the Rx FIFO is overrun.
5	RX_UNRUN	**Rx Underrun Interrupt** Asserts when the Rx FIFO is underrun.
4	RX_THRES	**Rx FIFO Below Threshold Interrupt** Asserts when the Rx FIFO is lower than the defined threshold.
3	TX_DMA_FAULT	**Tx DMA Fault Detected Interrupt** Asserts when a fault is detected in Tx DMA signals.
2	TX_OVRUN	**Tx FIFO Overrun Interrupt** Asserts when the Tx FIFO is overrun.
1	TX_UNRUN	**Tx FIFO Underrun Interrupt** Asserts when the Tx FIFO is underrun.
0	TX_THRES	**Tx FIFO Below Threshold Interrupt** Asserts when the FIFO is lower than the defined threshold.

10000A08 **INT_EN** **Interrupt Enable** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RX_INT3_EN	RX_INT2_EN	RX_INT1_EN	RX_INT0_EN	TX_INT3_EN	TX_INT2_EN	TX_INT1_EN	TX_INT0_EN
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	RX_INT3_EN	**INT_STATUS[7] Enable** Enables the Rx DMA Fault Detected Interrupt. This interrupt asserts when a fault is detected in Rx DMA signals.
6	RX_INT2_EN	**INT_STATUS[6] Enable** Enables the Rx Overrun Interrupt. This interrupt asserts when the Rx FIFO is overrun.
5	RX_INT1_EN	**INT_STATUS[5] Enable**

Bit(s)	Name	Description
		Enables the Rx Underrun Interrupt. This interrupt asserts when the Rx FIFO is underrun.
4	RX_INT0_EN	**INT_STATUS[4] Enable** Enables the Rx FIFO Below Threshold Interrupt. This interrupt asserts when the Rx FIFO is lower than the defined threshold.
3	TX_INT3_EN	**INT_STATUS[3] Enable** Enables the Tx DMA Fault Detected Interrupt. This interrupt asserts when a fault is detected in Tx DMA signals.
2	TX_INT2_EN	**INT_STATUS[2] Enable** Enables the Tx FIFO Overrun Interrupt. This interrupt asserts when the Tx FIFO is overrun.
1	TX_INT1_EN	**INT_STATUS[1] Enable** Enables the Tx FIFO Underrun Interrupt. This interrupt asserts when the Tx FIFO is underrun.
0	TX_INT0_EN	**INT_STATUS[0] Enable** Enables the Tx FIFO Below Threshold Interrupt. This interrupt asserts when the FIFO is lower than the defined threshold.

10000A0C **FF_STATUS** **FIFO Status** **00000010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				RX_AVCNT								TX_EPCNT				
Type				RO								RO				
Reset				0	0	0	0	0				1	0	0	0	0

Bit(s)	Name	Description
12:8	RX_AVCNT	**Rx FIFO Available Space Count** Counts the available space for reads in Rx FIFO. (unit: word)
4:0	TX_EPCNT	**Tx FIFO Available Space Count** Counts the available space for writes in Tx FIFO. (unit: word)

10000A10 **TX_FIFO_WREG** **Transmit FIFO Write to Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TX_FIFO_WDATA[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TX_FIFO_WDATA[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_FIFO_WDATA	**Tx FIFO Write Data Buffer** Buffers data to be written to the Tx FIFO.

10000A14 **RX_FIFO_RREG** **Receive FIFO Read Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_FIFO_RDATA[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_FIFO_RDATA[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RX_FIFO_RDATA	**Rx FIFO Read Data Buffer** Buffers data read from the Rx FIFO.

10000A18 **I2S_CFG1** **I2S Configuration 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LBK_EN	EXT_LBK_EN														
Type	RW	RW														
Reset	0	0														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																I2S_FMT
Type																RW
Reset																0

Bit(s)	Name	Description
31	LBK_EN	**Enables loopback mode.** 0: Normal mode 1: Loopback mode ASYNC_TXFIFIO -> Tx -> Rx -> ASYNC_RXFIFIO
30	EXT_LBK_EN	**Enables external loopback.** 0: Normal mode 1: Enables external loop back. External A/D -> Rx -> Tx -> External D/A
0	I2S_FMT	**I2S audio data format** 0: i2s mode 1: left-justified mode

10000A20 **DIVCOMP_CFG** **Integer Part of the Dividor Register 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CLK_EN															
Type	RW															
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								DIVCOMP								
Type								RW								
Reset								0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	CLK_EN	**Enables setting of the I2S clock based on DIVCOMP and DIVINT parameters.** 0: Disable 1: Enable
8:0	DIVCOMP	**A parameter in an equation which determines FREQOUT. See DIVINT_CFG.**

10000A28 **DIVINT_CFG** **Integer Part of the Dividor Register 2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DIVINT									
Type							RW									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
9:0	DIVINT	**Integer Divider** A parameter in an equation which determines FREQOUT: FREQOUT = FREQIN *(1/2) * {1 / [DIVINT+DIVCOMP/(512)]} FREQIN is always fixed to 480 MHz.

2.12 SPI Controller

2.12.1 Features

- Supports up to 2 SPI master operations
- Programmable clock polarity
- Programmable interface clock rate
- Programmable bit ordering
- Firmware-controlled SPI enable
- Programmable payload (address + data) length
- Supports 1/2/4 multi-IO SPI flash memory
- Supports command/user mode operation
- Supports SPI direct access
- Extends the addressable range from 24 bits to 32 bits for memory size larger than 128 Mb.

2.12.2 Block Diagram

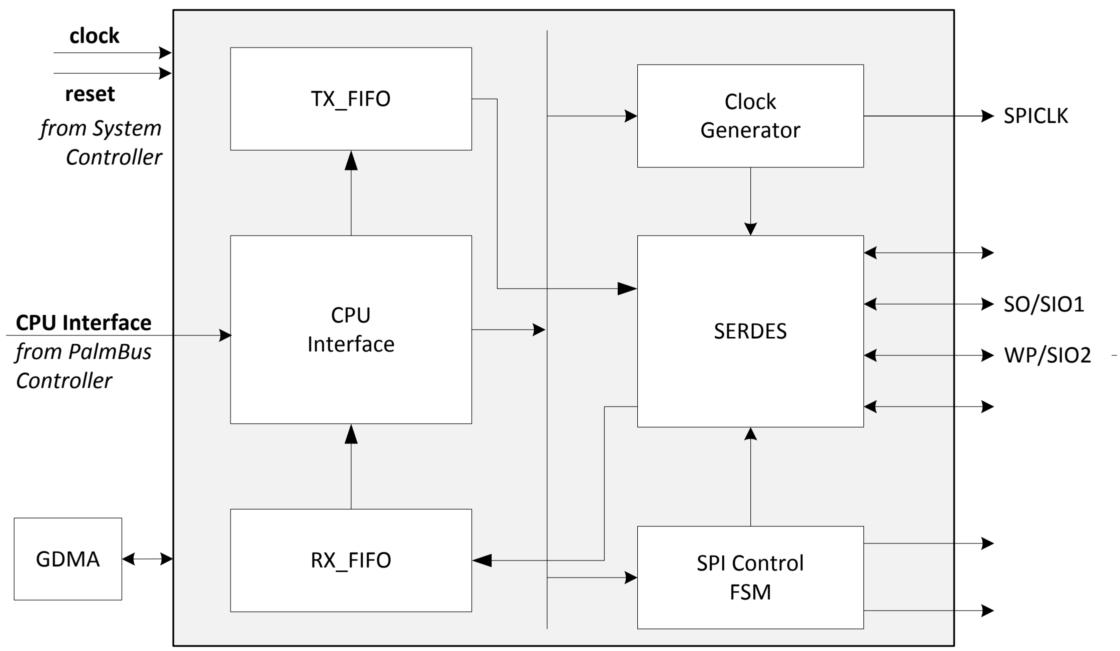


Figure 2-7 SPI Controller Block Diagram

2.12.3 Registers

SPI Changes LOG

Revision	Date	Author	Change Log
0.1	2012/8/29	Lancelot	Initialization
0.2	2012/11/6	Lancelot	1. Remove 0x38 SW_RST 2. Add CS_POLAR at 0x38
0.3	2012/11/23	Lancelot	Fix default value

Module name: SPI Base address: (+10000B00h)

Address	Name	Width	Register Function
10000B00	SPI_TRANS	32	SPI transaction control/status register
10000B04	SPI_OP_ADDR	32	SPI opcode/address register
10000B08	SPI_DIDO_0	32	SPI DI/DO data #0 register
10000B0C	SPI_DIDO_1	32	SPI DI/DO data #1 register
10000B10	SPI_DIDO_2	32	SPI DI/DO data #2 register
10000B14	SPI_DIDO_3	32	SPI DI/DO data #3 register
10000B18	SPI_DIDO_4	32	SPI DI/DO data #4 register
10000B1C	SPI_DIDO_5	32	SPI DI/DO data #5 register
10000B20	SPI_DIDO_6	32	SPI DI/DO data #6 register
10000B24	SPI_DIDO_7	32	SPI DI/DO data #7 register
10000B28	SPI_MASTER	32	SPI master mode register
10000B2C	SPI_MORE_BUF	32	SPI more buf control register
10000B30	SPI_QUEUE_CTL	32	SPI flash queue control register
10000B34	SPI_STATUS	32	SPI controller status register
10000B38	SPI_CS_POLAR	32	SPI chip select polarity
10000B3C	SPI_SPACE	32	SPI flash space control register

10000B00 SPI_TRANS SPI transaction control/status register 00160001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	spi_addr_ext								Reserved0			spi_addr_size		Reserved1		spi_master_busy
Type	RW								RO			RW		RO		RO
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved2							spi_master_start	miso_byte_cnt				mosi_byte_cnt			
Type	RO							WO	RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:24	spi_addr_ext	**SPI address extention** Address extenstion for 32-bit SPI address size. Usually this field specifies the first byte of the address phase to transmit to SPI device when more_buf_mode = 0 and spi_addr_size = 3. And spi_addr[31:24], spi_addr[23:16], and spi_addr[15:0] are respectively the second, third and fourth byte of the address phase
20:19	spi_addr_size	**SPI address size.**

Bit(s)	Name	Description
		0: reserved. 1: spi_addr[15:0] of SPI DI data register are valid (16-bit size). 2: spi_addr[23:0] of SPI DI data register are valid (24-bit size). 3: {spi_addr_ext[7:0], spi_addr[23:0]} of SPI DI data register are valid (32-bit size) Note: The spi_addr_size is valid only when more_buf_mode = 0.
16	spi_master_busy	**Transaction busy indication (Read-only). Writes to this bit are ignored.** 0: No SPI transaction is ongoing. Software may start a new SPI transaction by writing to the SPI transaction start bit within this register. 1: An SPI transaction presently is underway. Software must not try to start a new SPI transaction. Software may not alter the value of any field of the SPI master control registers.
8	spi_master_start	**SPI transaction start. Only writes to this field are meaningful, reads always return 0.** Writes: 0: No effect 1: Starts SPI transaction.
7:4	miso_byte_cnt	**SPI MISO (rx) byte count.** Determines the number of bytes received from the SPI device from the SPI opcode/address register and the SPI DI/DO data #0 register. Values of 0 ~ 8 are valid, other values are illegal. Note: The miso_byte_cnt is valid only when more_buf_mode = 0.
3:0	mosi_byte_cnt	**SPI MOSI (tx) byte count.** Determines the number of bytes transmitted from the SPI opcode/address register and the SPI DI/DO data #0 register to the SPI device. Values of 1 ~ 8 are valid, other values are illegal. Note: The mosi_byte_cnt is valid only when more_buf_mode = 0. The transmitted data sequence is as follows: spi_opcode, spi_addr (conditional) and d0_byte ~ d3_byte (conditional).

10000B04 **SPI_OP_ADDR** **SPI opcode/address register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**spi_addr[23:8]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**spi_addr[7:0]**								**spi_opcode**							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:8	spi_addr	**SPI address. Usually this field specifies the 24-bits address to transmit to the SPI device when more_buf_mode = 0.** 1: (16-bits SPI address size), spi_addr[23:16] is the 1st byte of the address phase and spi_addr[15:8] is the 2nd byte of the address phase. 2: (24-bits SPI address size), spi_addr[31:24] is the 1st byte of the address phase and spi_addr[23:16] is the 2nd byte of the address phase and spi_addr[15:8] is the 3rd byte of the address phase. 3: (32-bits SPI address size), spi_addr[31:24] is the 2nd byte of the address phase and spi_addr[23:16] is the 3rd byte of the address phase and spi_addr[15:8] is the 4th byte of the address phase Note: For SPI read transaction and more_buf_mode = 0 Field [15:8] is also used to store the 6-th byte of data read phase. Field [23:16] is also used to store the 7-th byte of data read phase. Field [31:24] is also used to store the 8-th byte of data read phase.

Bit(s)	Name	Description
7:0	spi_opcode	**SPI opcode. Usually this field specifies the 8-bits opcode (instruction) to transmit to the SPI device as the first byte of a SPI transaction when more_buf_mode = 0.** Note: For SPI read transaction and more_buf_mode = 0, this byte is also used to store the 5-th byte of data read phase according to the rx byte count miso_byte_cnt.

10000B08 **SPI_DIDO_0** **SPI DI/DO data #0 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**d3_byte**								**d2_byte**							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**d1_byte**								**d0_byte**							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B0C **SPI_DIDO_1** **SPI DI/DO data #1 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**d3_byte**								**d2_byte**							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**d1_byte**								**d0_byte**							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B10 **SPI_DIDO_2** **SPI DI/DO data #2 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**d3_byte**								**d2_byte**							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**d1_byte**								**d0_byte**							

Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B14 SPI_DIDO_3 SPI DI/DO data #3 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	d3_byte								d2_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	d1_byte								d0_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B18 SPI_DIDO_4 SPI DI/DO data #4 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	d3_byte								d2_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	d1_byte								d0_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B1C SPI_DIDO_5 SPI DI/DO data #5 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	d3_byte								d2_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	d1_byte								d0_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B20 SPI_DIDO_6 SPI DI/DO data #6 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	d3_byte								d2_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	d1_byte								d0_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B24 SPI_DIDO_7 SPI DI/DO data #7 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	d3_byte								d2_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	d1_byte								d0_byte							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	d3_byte	**The 4th data byte of data read/write phase.**
23:16	d2_byte	**The 3th data byte of data read/write phase.**
15:8	d1_byte	**The 2nd data byte of data read/write phase.**
7:0	d0_byte	**The 1st data byte of data read/write phase.**

10000B28 SPI_MASTER SPI master mode register 000D8880

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	rs_slave_sel			clk_	rs_clk_sel											

				mode												
Type	RW			RW	RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**cs_dsel_cnt**					**full_duplex**	**int_en**	**spi_start_sel**	**spi_prefetch**	**bidir_mode**	**cpha**	**cpol**	**lsb_first**	**more_buf_mode**	**serial_mode**	
Type	RW					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Reset	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:29	rs_slave_sel	**select SPI device** 0: select SPI device 0 (default is flash) 1: select SPI device 1 ... 7: select SPI device 7
28	clk_mode	**This register is used to specify that period of SCLK HIGH is longer or period of SCLK LOW is longer when clock divisor(clk_sel) is odd.** 0: period of SCLK LOW is longer. 1: period of SCLL HIGH is longer.
27:16	rs_clk_sel	**Register Space SPI clock frequency select.** 0: SPI clock frequency is hclk/2. (50% duty cycle, duty cycle is the ratio of the output high time to the total cycle time) 1: SPI clock frequency is hclk/3. (33.33% or 66.67% duty cycle) 2: SPI clock frequency is hclk/4. (50% duty cycle) 3: SPI clock frequency is hclk/5. (40% or 60% duty cycle) 4095: SPI clock frequency is hclk/4097.
15:11	cs_dsel_cnt	**De-select time of SPI chip select is configured to occupy the number of cycles of AHB clock**
10	full_duplex	**Full duplex or half duplex mode.** 0: half duplex mode. 1: full duplex mode. Full duplex timing diagram Note: The full_duplex is valid only when more_buf_mode = 1. The transmission is always as half duplex when more_buf_mode = 0;
9	int_en	**Interrupt enable.** 0: disable SPI interrupt. 1: enable SPI interrupt.
8	spi_start_sel	**The interval between spi_cs_n and spi_sclk.** 0: 3 clk 1: 6 clk
7	spi_prefetch	**SPI pre-fetch buffer enable** 0: disable pre-fetch buffer. 1: enable pre-fetch buffer.
6	bidir_mode	**Bi-direction mode. In this mode, the SPI uses only one serial data pin for interface with external devices. The MOSI pin becomes the serial data I/O pin for the SPI transaction and MISO pin is not used. Bi-direction mode is used for the application with only 1 bi-direction serial pin for SPI transaction.** 0: normal mode (both MOSI and MISO pins are used). 1: bi-direction mode (only MOSI pin is used). SPI host controller must operate in half duplex mode if bidir_mode = 1. Note: The bidir_mode is valid only when more_buf_mode = 1.
5	cpha	**(CPHA, clock phase). Initial SPI clock phase for SPI transaction.** There are four SPI modes used to latch data. These SPI modes latch data in

Bit(s)	Name	Description
		one of four ways, and are defined by the logic state combinations of the CLK Polarity (CPOL) in relation to the CLK Phase (CPHA). The valid logic combinations identify and determine the SPI modes supported by the SPI device. SPI mode At CPOL=0 the base value of the clock is zero For CPHA=0 (mode 0), data is read on the clock's rising edge and data is changed on a falling edge. For CPHA=1 (mode 1), data is read on the clock's falling edge and data is changed on a rising edge. At CPOL=1 the base value of the clock is one (inversion of CPOL=0) For CPHA=0 (mode 2), data is read on clock's falling edge and data is changed on a rising edge. For CPHA=1 (mode 3), data is read on clock's rising edge and data is changed on a falling edge.
4	cpol	**cpol (CPOL, clock polarity). Initial SPI clock polarity for SPI transaction.**
3	lsb_first	**0: MSB(most significant bit) is transferred first for SPI transaction.** 1: LSB(least significant bit) is transferred first for SPI transaction.
2	more_buf_mode	**Select 2 words buffer or 8 words buffer for SPI transaction.** 0: SPI transfer data buffer size is only 2 words. In this mode, SPI DI/DO data #0 register and SPI opcode/address register are the data buffer for SPI transaction. And, SPI master follows mosi_byte_cnt and miso_byte_cnt to complete the transmission and reception, respectively. This kind of transaction must operate in half duplex mode. 1: SPI transfer data buffer size is 8 words. In this mode, SPI opcode/address register are the data buffer for SPI transaction and follows cmd_bit_cnt to complete the transaction. SPI DI/DO data #0~#7 register are the data buffer for SPI transaction and follows do_bit_cnt and di_bit_cnt to complete the transmission and reception, respectively. In half duplex mode, transmitted data are loaded from SPI opcode/address register and SPI DI/DO data #0~#7 registers. And, the received data will overwrite the SPI DI/DO data #0~#7 registers. In full duplex mode, SPI DI/DO data #0~#3 registers are used for transmission and SPI DI/DO #4~#7 registers are used for receipt.
1:0	serial_mode	**This mode is designed for Winbond SPI flash W25Q80/16/32 and W25X10/20/40/80/16/32/64 series.** 0: standard serial. 1: dual serial. 2: quad serial. 3: reserved. Note: The serial_mode is valid only when more_buf_mode = 0. The transaction mode is always as standard serial when more_buf_mode = 1.

10000B2C **SPI_MORE_BUF** **SPI more buf control register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved0		cmd_bit_cnt						Reserved1			miso_bit_cnt[8:4]				
Type	RO		RW						RO			RW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	miso_bit_cnt[3:0]				Reserved2			mosi_bit_cnt								
Type	RW				RO			RW								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
29:24	cmd_bit_cnt	**SPI command phase MOSI (tx) bit count. Determines the number of command bits transmitted from the SPI opcode/address register to the SPI device. Values of 0 ~ 32 are valid, but other values are illegal.** Note: The cmd_bit_cnt is valid only when more_buf_mode = 1 and the SPI opcode/address register is treated as a command register.
20:12	miso_bit_cnt	**SPI data phase MISO (rx) bit count. Determines the number of bits received from the SPI device into the SPI DI/DO data #0~#7 register. Values of 0 ~ 256 are valid, but other values are illegal. Maximum value is 256 for half duplex mode and 128 for full duplex mode. Please note that do_bit_cnt must be equal to di_bit_cnt in full duplex mode.** Note: The miso_bit_cnt is valid only when more_buf_mode = 1.
8:0	mosi_bit_cnt	**SPI data phase MOSI (tx) bit count. Determines the number of data bits transmitted from the SPI DI/DO data #0~#7 register to the SPI device. Values of 0 ~ 256 are valid, but other values are illegal. Maximum value is 256 for half duplex mode and 128 for full duplex mode.** Note: The mosi_bit_cnt is valid only when more_buf_mode = 1.

10000B30 **SPI_QUEUE_CTL** **SPI flash queue control register** **00000E40**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	fs_page_sel						Reserved0[12:3]									
Type	RW						RO									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved0[2:0]			fs_busy	fs_addr_size_r		fs_addr_size		fs_di_ph_byc				Reserved1	fast_spi_sel		
Type	RO			RO	RO		RW		RW				RO	RW		
Reset	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0

Bit(s)	Name	Description
31:26	fs_page_sel	**Flash Space Page Selection.** 0: (Page 0 space) 0x0000_0000 - 0x03ff_ffff 1: (Page 1 space) 0x0400_0000 - 0x07ff_ffff ... 63: (Page 63 space) 0xffc0_0000 - 0xffff_ffff
12	fs_busy	**Transaction busy indication (Read-only) in flash space. Writes to this bit are ignored.** 0: No SPI flash space access is ongoing. Software may change the configuration related to flash space. 1: SPI flash space access presently is underway. Software may not alter the configuration related to flash space.
11:10	fs_addr_size_r	**Latched fs_addr_size indication from internal spimc logic**
9:8	fs_addr_size	**SPI address. This field specifies the 24-bits/16-bits address to transmit to the SPI device for SPI Flash Space Read operation only.** 0: 25-bit SPI address size 1: 16-bit SPI address size Reserved. 2: 24-bit SPI address size (default for 3B SPI flash) 3: 26-bit SPI address size (default for 4B SPI flash) If the change of the fs_addr_size is needed, the sequence below must be followed. Otherwise, the new fs_addr_size configuration will not be updated to the internal spimc logic . Step 1: Set new fs_addr_size. Step 2: Transmit mode change command (ex. En4B or Ex4B of MX25L25635E) Note: 1. The value fs_addr_size is not valid in Register Space.

Bit(s)	Name	Description
		2. The Spimc now only supports 3-Byte mode (24 bits) and 4-Byte mode (25 or 26 bits) switch.
7:4	fs_di_ph_byc	**Determines the number of data bytes transmitted from the SPI master controller to the SPI device for SPI Flash Space Read operation. This field is similar to mosi_byte_cnt in STCSR but is used for setting of flash space access control path.** Note: this field should (if fs_addr_size_r = 2, 24-bit fs_addr_size) = 4 (OP + ADDR) if fast_spi_sel = 0 (0x03) = 5 (OP + ADDR + dummy) if fast_spi_sel = 1 (0x0b) = 5 (OP + ADDR + dummy) if fast_spi_sel = 2 (0x3b) = 5 (OP + ADDR + M7-0) if fast_spi_sel = 3 (0xbb) = 5 (OP + ADDR + dummy) if fast_spi_sel = 4 (0x6b) = 7 (OP + ADDR + M7-0 + dummy) if fast_spi_sel = 5 (0xeb) = 5 (OP + ADDR + M7-0) if fast_spi_sel = 6 (0xe3) (if fs_addr_size_r = 0 or 3, 25 or 26-bit fs_addr_size) = 5 (OP + ADDR) if fast_spi_sel = 0 (0x03) = 6 (OP + ADDR + dummy) if fast_spi_sel = 1 (0x0b) = 6 (OP + ADDR + dummy) if fast_spi_sel = 2 (0x3b) = 6 (OP + ADDR + M7-0) if fast_spi_sel = 3 (0xbb) = 6 (OP + ADDR + dummy) if fast_spi_sel = 4 (0x6b) = 8 (OP + ADDR + M7-0 + dummy) if fast_spi_sel = 5 (0xeb) = 6 (OP + ADDR + M7-0) if fast_spi_sel = 6 (0xe3)
2:0	fast_spi_sel	**Select SPI flash read instruction for Flash Space** 0: standard read data instruction (0x03). 1: standard fast read data instruction (0x0b). 2: fast read dual output instruction defined in Winbond W25Qxx series SPI flash (0x03b). 3: fast read dual I/O instruction defined in Winbond W25Qxx series SPI flash (0xbb). 4: fast read quad output instruction defined in Winbond W25Qxx series SPI flash (0x6b). 5: fast read quad I/O instruction defined in Winbond W25Qxx series SPI flash (0xeb). 6: burst read quad I/O instruction defined in Winbond W25Qxx series SPI flash (0xe3). Note: serial_mode and more_buf_mode are don't care for this flash space access control path.

10000B34 SPI_STATUS SPI controller status register 00000030

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved0[25:10]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved0[9:0]										spi_flash_mode		Reserved1			spi_ok
Type	RO										RO		RO			RC
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit(s)	Name	Description
5:4	spi_flash_mode	**0: no SPI flash.** 1: standard SPI flash. 2: specific SPI flash with dual interface capability. 3: specific SPI flash with quad interface capability.

Bit(s)	Name	Description
0	spi_ok	**When SPI transaction complete, SPI master controller will set this bit and assert SPI interrupt to notify software. Reading this register will clear this bit and de-assert SPI interrupt.**

10000B38 **SPI_CS_POLAR** **SPI chip select polarity** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**cs_polar**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	cs_polar	**Chip select default polarity** set cs_polar[n]=1'b0 for cs[n] low active (SPI Flash) set cs_polar[n]=1'b1 for cs[n] high active

10000B3C **SPI_SPACE** **SPI flash space control register** **00000030**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**Reserved[16:1]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved[0:0]**	**fs_slave_sel**			**fs_clk_sel**											
Type	RO	RW			RW											
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit(s)	Name	Description
14:12	fs_slave_sel	**(Flash Space Slave Select)** 0: select SPI device #0. (default is flash) 1: select SPI device #1. ... 7: select SPI device #7.
11:0	fs_clk_sel	**Flash Space SPI clock frequency select.** 0: SPI clock frequency is hclk/2. (50% duty cycle, duty cycle is the ratio of the output high time to the total cycle time) 1: SPI clock frequency is hclk/3. (33.33% or 66.67% duty cycle) 2: SPI clock frequency is hclk/4. (50% duty cycle) 3: SPI clock frequency is hclk/5. (40% or 60% duty cycle) 4095: SPI clock frequency is hclk/4097.

2.13 UART Lite

2.13.1 Features

- 2-pin UART
- 16550-compatible register set, except for Divisor Latch register
- 5-8 data bits
- 1-2 stop bits (1 or 2 stop bits are supported with 5 data bits)
- Even, odd, stick or no parity
- All standard baud rates up to 345600 b/s
- 16-byte receive buffer
- 16-byte transmit buffer
- Receive buffer threshold interrupt
- Transmit buffer threshold interrupt
- False start bit detection in asynchronous mode
- Internal diagnostic capabilities
- Break simulation
- Loop-back control for communications link fault isolation

2.13.2 Registers

n = 1; for uart1 only.

UARTn+0000h RX Buffer Register — UARTn_RBR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RBR[7:0]**							
Type									RO							

RBR RX Buffer Register. Read-only register. The received data can be read by accessing this register.
Modified when LCR[7] = 0.

UARTn+0000h TX Holding Register — UARTn_THR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**THR[7:0]**							
Type									WO							

THR TX Holding Register. Write-only register. The data to be transmitted is written to this register, and then sent to the PC via serial communication.
Modified when LCR[7] = 0.

UARTn+0004h Interrupt Enable Register — UARTn_IER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**CTSI**	**RTSI**	**XOFFI**	**X**	**EDSSI**	**ELSI**	**ETBEI**	**ERBFI**
Type									R/W							
Reset									0							

IER By storing a '1' to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt is disabled.
IER[3:0] are modified when LCR[7] = 0.
IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.

CTSI Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line.
Note: This interrupt is only enabled when hardware flow control is enabled.

0 Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.

1 Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line.

RTSI Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line.

Note: This interrupt is only enabled when hardware flow control is enabled.

0 Unmask an interrupt that is generated when a rising edge is detected on the RTS modem control line.

1 Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line.

XOFFI Masks an interrupt that is generated when an XOFF character is received.

Note: This interrupt is only enabled when software flow control is enabled.

0 Unmask an interrupt that is generated when an XOFF character is received.

1 Mask an interrupt that is generated when an XOFF character is received.

EDSSI When set ("1"), an interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

0 No interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

1 An interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

ELSI When set ("1"), an interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.

0 No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.

1 An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.

ETBEI When set ("1"), an interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO

have been reduced to its Trigger Level.

0 No interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level.

1 An interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level

ERBFI When set ("1"), an interrupt is generated if the RX Buffer contains data.

0 No interrupt is generated if the RX Buffer contains data.

1 An interrupt is generated if the RX Buffer contains data.

UARTn+0008h Interrupt Identification Register UARTn_IIR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID4	ID3	ID2	ID1	ID0	NINT
Type									RO							
Reset									0	0	0	0	0	0	0	1

IIR Identify if there are pending interrupts; ID4 and ID3 are presented only when EFR[4] = 1.

The following table gives the IIR[5:0] codes associated with the possible interrupts:

IIR[5:0]	Priority Level	Interrupt	Source
000001	-	No interrupt pending	
000110	1	Line Status Interrupt	BI, FE, PE or OE set in LSR
000100	2	RX Data Received	RX Data received or RX Trigger Level reached.
001100	2	RX Data Timeout	Timeout on character in RX FIFO.
000010	3	TX Holding Register Empty	TX Holding Register empty or TX FIFO Trigger Level reached.
000000	4	Modem Status change	DDCD, TERI, DDSR or DCTS set in MSR

010000	5	Software Flow Control	XOFF Character received
100000	6	Hardware Flow Control	CTS or RTS Rising Edge

Table 1 The IIR[5:0] codes associated with the possible interrupts

Line Status Interrupt: A RX Line Status Interrupt (IIR[5:0`] == 000110b) is generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the Line Status Register.
RX Data Received Interrupt: A RX Received interrupt (IER[5:0] == 000100b) is generated if EFRBI (IER[0]) is set and either RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached. The interrupt is cleared by reading the RX Buffer Register or the RX FIFO (if enabled).
RX Data Timeout Interrupt:
When virtual FIFO mode is disabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO contains at least one character;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register, or on a CPU read from the RX FIFO.
The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading RX FIFO.
When virtual FIFO mode is enabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO is empty;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register.
RX Holding Register Empty Interrupt: A TX Holding Register Empty Interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX Holding Register or, if FIFOs are enabled, the TX FIFO becomes empty. The interrupt is cleared by writing to the TX Holding Register or TX FIFO if FIFO enabled.
Modem Status Change Interrupt: A Modem Status Change Interrupt (IIR[5:0] = 000000b) is generated if EDSSI (IER[3]) is set and either DDCD, TERI, DDSR or DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the Modem Status Register.
Software Flow Control Interrupt: A Software Flow Control Interrupt (IIR[5:0] = 010000b) is generated if Software Flow Control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the Interrupt Identification Register.
Hardware Flow Control Interrupt: A Hardware Flow Control Interrupt (IER[5:0] = 100000b) is generated if Hardware Flow Control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS Modem Control line. The interrupt is cleared by reading the Interrupt Identification Register.

UARTn+0008h FIFO Control Register UARTn_FCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1	RFTL0	TFTL1	TFTL0	DMA1	CLRT	CLRR	FIFOE
Type									WO							

FCR FCR is used to control the trigger levels of the FIFOs, or flush the FIFOs.
FCR[7:6] is modified when LCR != BFh
FCR[5:4] is modified when LCR != BFh & EFR[4] = 1

FCR[4:0] is modified when LCR != BFh

FCR[7:6] RX FIFO trigger threshold

0 1

1 6

2 12

3 **RXTRIG**

FCR[5:4] TX FIFO trigger threshold

0 1

1 4

2 8

3 14 (FIFOSIZE - 2)

DMA1 This bit determines the DMA mode, which the TXRDY and RXRDY pins support. TXRDY and RXRDY act to support single-byte transfers between the UART and memory (DMA mode 0) or multiple byte transfers (DMA mode1). Note that this bit has no effect unless the FIFOE bit is set as well

0 The device operates in DMA Mode 0.

1 The device operates in DMA Mode 1.

TXRDY – mode0: Goes active (low) when the TX FIFO or the TX Holding Register is empty. Becomes inactive when a byte is written to the Transmit channel.

TXRDY – mode1: Goes active (low) when there are no characters in the TX FIFO. Becomes inactive when the TX FIFO is full.

RXRDY – mode0: Becomes active (low) when at least one character is in the RX FIFO or the RX Buffer Register is full. Becomes inactive when there are no more characters in the RX FIFO or RX Buffer register.

RXRDY – mode1: Becomes active (low) when the RX FIFO Trigger Level is reached or an RX FIFO Character Timeout occurs. Goes inactive when the RX FIFO is empty.

CLRT Clear Transmit FIFO. This bit is self-clearing.

0 Leave TX FIFO intact.

1 Clear all the bytes in the TX FIFO.

CLRR Clear Receive FIFO. This bit is self-clearing.

0 Leave RX FIFO intact.

1 Clear all the bytes in the RX FIFO.

FIFOE FIFO Enabled. This bit must be set to 1 for any of the other bits in the registers to have any effect.

0 Disable both the RX and TX FIFOs.

1 Enable both the RX and TX FIFOs.

UARTn+000Ch Line Control Register UARTn_LCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLAB**	**SB**	**SP**	**EPS**	**PEN**	**STB**	**WLS1**	**WLS0**
Type									R/W							
Reset									0	0	0	0	0	0	0	0

LCR Line Control Register. Determines characteristics of serial communication signals.

Modified when LCR[7] = 0.

DLAB Divisor Latch Access Bit.

0 The RX and TX Registers are read/written at Address 0 and the IER register is read/written at Address 4.

1 The Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.

SB Set Break

0 No effect

1 SOUT signal is forced into the "0" state.

SP Stick Parity

0 No effect.

1 The Parity bit is forced into a defined state, depending on the states of EPS and PEN:
If EPS=1 & PEN=1, the Parity bit is set and checked = 0.
If EPS=0 & PEN=1, the Parity bit is set and checked = 1.

EPS Even Parity Select

0 When EPS=0, an odd number of ones is sent and checked.

1 When EPS=1, an even number of ones is sent and checked.

PEN Parity Enable

0 The Parity is neither transmitted nor checked.

1 The Parity is transmitted and checked.

STB Number of STOP bits

0 One STOP bit is always added.

1 Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.

WLS1, 0 Word Length Select.

0 5 bits

1 6 bits

2 7 bits

3 8 bits

UARTn+0010h Modem Control Register UARTn_MCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF STATUS		X	DCM_EN	OUT2	OUT1	RTS	DTR
Type									R/W							
Reset									0		0	0	0	0	0	0

MCR Modem Control Register. Control interface signals of the UART.
MCR[4:0] are modified when LCR[7] = 0,
MCR[7:6] are modified when LCR[7] = 0 & EFR[4] = 1.

XOFF Status This is a read-only bit.

0 When an XON character is received.

1 When an XOFF character is received.

DCM_EN UART DCM function enable bit

0 UART DCM is disabled.

1 UART DCM is enabled.

OUT2 Controls the state of the output NOUT2, even in loop mode.

0 NOUT2=1.

1 NOUT2=0.

OUT1 Controls the state of the output NOUT1, even in loop mode.

0 NOUT1=1.

1 NOUT1=0.

RTS Controls the state of the output NRTS, even in loop mode.

0 NRTS=1.

1 NRTS=0.

DTR Control the state of the output NDTR, even in loop mode.

0 NDTR=1.

1 NDTR=0.

UARTn+0014h Line Status Register — UARTn_LSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
Type									R/W							
Reset									0	1	1	0	0	0	0	0

LSR Line Status Register.
Modified when LCR[7] = 0.

FIFOERR RX FIFO Error Indicator.

0 No PE, FE, BI set in the RX FIFO.

1 Set to 1 when there is at least one PE, FE or BI in the RX FIFO.

TEMT TX Holding Register (or TX FIFO) and the TX Shift Register are empty.

0 Empty conditions below are not met.

1 If FIFOs are enabled, the bit is set whenever the TX FIFO and the TX Shift Register are empty. If FIFOs are disabled, the bit is set whenever TX Holding Register and TX Shift Register are empty.

THRE Indicates if there is room for TX Holding Register or TX FIFO is reduced to its Trigger Level.

0 **Reset whenever the contents of the TX FIFO are more than its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is not empty(FIFOs are disabled).**

1 Set whenever the contents of the TX FIFO are reduced to its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is empty and ready to accept new data (FIFOs are disabled).

BI Break Interrupt.

0 Reset by the CPU reading this register

1 If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits).
If the FIFOs are enabled, this error is associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: the next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.

FE Framing Error.

0 Reset by the CPU reading this register

1 If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is the next to be read.

PE Parity Error

0 Reset by the CPU reading this register

1 If the FIFOs are disabled, this bit is set if the received data did not have a valid parity bit. If the FIFOs are enabled, the state of this bit is revealed when the referred byte is the next to be read.

OE Overrun Error.

0 Reset by the CPU reading this register.

1 If the FIFOs are disabled, this bit is set if the RX Buffer was not read by the CPU before new data from the RX Shift Register overwrote the previous contents.
If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX Shift Register becomes full. OE is set as soon as this happens. The character in the Shift Register is then overwritten, but not transferred to the FIFO.

DR Data Ready.

0 Cleared by the CPU reading the RX Buffer or by reading all the FIFO bytes.

1 Set by the RX Buffer becoming full or by a byte being transferred into the FIFO.

UARTn+0018h Modem Status Register UARTn_MSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DCD**	**RI**	**DSR**	**CTS**	**DDCD**	**TERI**	**DDSR**	**DCTS**
Type									R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset									Input	Input	Input	Input	0	0	0	0

Note: After a reset, D4-D7 are inputs. A modem status interrupt can be cleared by writing '0' or set by writing '1' to this register. D0-D3 can be written to.
Modified when LCR[7] = 0.

MSR Modem Status Register

DCD Data Carry Detect.

When Loop = "0", this value is the complement of the NDCD input signal.
When Loop = "1", this value is equal to the OUT2 bit in the Modem Control Register.

RI Ring Indicator.

When Loop = "0", this value is the complement of the NRI input signal.
When Loop = "1", this value is equal to the OUT1 bit in the Modem Control Register.

DSR Data Set Ready

When Loop = "0", this value is the complement of the NDSR input signal.
When Loop = "1", this value is equal to the DTR bit in the Modem Control Register.

CTS Clear To Send.

When Loop = "0", this value is the complement of the NCTS input signal.
When Loop = "1", this value is equal to the RTS bit in the Modem Control Register.

DDCD Delta Data Carry Detect.

0 The state of DCD has not changed since the Modem Status Register was last read

1 Set if the state of DCD has changed since the Modem Status Register was last read.

TERI Trailing Edge Ring Indicator

0 The NRI input does not change since this register was last read.

1 Set if the NRI input changes from "0" to "1" since this register was last read.

DDSR Delta Data Set Ready

0 Cleared if the state of DSR has not changed since this register was last read.

1 Set if the state of DSR has changed since this register was last read.

DCTS Delta Clear To Send

0 Cleared if the state of CTS has not changed since this register was last read.
1 Set if the state of CTS has changed since this register was last read.

UARTn+001Ch Scratch Register UARTn_SCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SCR[7:0]**							
Type									R/W							

A general purpose read/write register. After reset, its value is un-defined.
Modified when LCR[7] = 0.

UARTn+0000h Divisor Latch (LS) UARTn_DLL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLL[7:0]**							
Type									R/W							
Reset									1							

UARTn+0004h Divisor Latch (MS) UARTn_DLM

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLL[7:0]**							
Type									R/W							
Reset									0							

Note: DLL & DLM can only be updated if DLAB is set (“1”).. Note too that division by 1 generates a BAUD signal that is constantly high.
Modified when LCR[7] = 1.
The table below shows the divisor needed to generate a given baud rate from CLK inputs of 13, 26 MHz and 52 MHz. The effective clock enable generated is 16 x the required baud rate.

BAUD	**13MHz**	**26MHz**	**52MHz**
110	7386	14773	29545
300	2708	5417	10833
1200	677	1354	2708
2400	338	677	1354
4800	169	339	677
9600	85	169	339
19200	42	85	169
38400	21	42	85
57600	14	28	56
115200	6	14	28

Table 2 Divisor needed to generate a given baud rate

UARTn+0008h Enhanced Feature Register UARTn_EFR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**AUTO CTS**	**AUTO RTS**	**D5**	**ENABLE -E**	**SW FLOW CONT[3:0]**			
Type									R/W	R/W	R/W	R/W	R/W			
Reset									0	0	0	0	0			

*NOTE: Only when LCR=BF’h

Auto CTS Enables hardware transmission flow control
- **0** Disabled.
- **1** Enabled.

Auto RTS Enables hardware reception flow control
- **0** Disabled.
- **1** Enabled.

Enable-E Enable enhancement features.
- **0** Disabled.
- **1** Enabled.

CONT[3:0] Software flow control bits.
- **00xx** No TX Flow Control
- **10xx** Transmit XON1/XOFF1 as flow control bytes
- **01xx** Transmit XON2/XOFF2 as flow control bytes
- **11xx** Transmit XON1 & XON2 and XOFF1 & XOFF2 as flow control words
- **xx00** No RX Flow Control
- **xx10** Receive XON1/XOFF1 as flow control bytes
- **xx01** Receive XON2/XOFF2 as flow control bytes
- **xx11** Receive XON1 & XON2 and XOFF1 & XOFF2 as flow control words

UARTn+0010h XON1 — UARTn_XON1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON1[7:0]							
Type									R/W							
Reset									0							

UARTn+0014h XON2 — UARTn_XON2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON2[7:0]							
Type									R/W							
Reset									0							

UARTn+0018h XOFF1 — UARTn_XOFF1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF1[7:0]							
Type									R/W							
Reset									0							

UARTn+001Ch XOFF2 — UARTn_XOFF2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF2[7:0]							
Type									R/W							
Reset									0							

UARTn+0024h HIGH SPEED UART — UARTn_HIGHSPEED

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SPEED [1:0]	
Type															R/W	
Reset															0	

SPEED UART sample counter base

- **0** based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL}
- **1** based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL}
- **2** based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL}
- **3** based on sampe_count * baud_pulse, baud_rate = system clock frequency / sampe_count

When HIGHSPEED=3, the value (A * B) means ({DLM, DLL} * SAMPLE_COUNT).
When the Baudrate is more than 115200, it will be more accurate if we set HIGHSPEED=3.
The table below shows the divisor needed to generate a given baud rate from CLK inputs of 13M Hz based on different HIGHSPEED value.

BAUD	HIGHSPEED = 0	HIGHSPEED = 1	HIGHSPEED = 2	HIGHSPEED = 3
110	7386	14773	29545	7386 * 16
300	2708	7386	14773	2708 * 16
1200	677	2708	7386	677 * 16
2400	338	677	2708	338 * 16
4800	169	338	677	169 * 16
9600	85	169	338	85 * 16
19200	42	85	169	9 * 75
38400	21	42	85	13 * 26
57600	14	21	42	8 * 28
115200	7	14	21	4 * 28
230400	*	7	14	2 * 28
460800	*	*	7	1 * 28
921600	*	*	*	1 * 14

Table 3 *Divisor needed to generate a given baud rate from 13MHz based on different HIGHSPEED value*

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 26 MHz based on different HIGHSPEED value.

BAUD	HIGHSPEED = 0	HIGHSPEED = 1	HIGHSPEED = 2	HIGHSPEED = 3
110	14773	29545	59091	7386 * 32
300	5417	14773	29545	2708 * 32
1200	1354	5417	14773	677 * 32
2400	677	1354	5417	338 * 32
4800	339	677	1354	169 * 32
9600	169	339	667	85 * 32
19200	85	169	339	18 * 75
38400	42	85	169	26 * 26

57600	28	42	85	16 * 28
115200	14	28	42	8 * 28
230400	7	14	28	4 * 28
460800	*	7	14	2 * 28
921600	*	*	7	1 * 28

Table 4 *Divisor needed to generate a given baud rate from 26 MHz based on different HIGHSPEED value*

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 52MHz based on different HIGHSPEED value.

BAUD	HIGHSPEED = 0	HIGHSPEED = 1	HIGHSPEED = 2	HIGHSPEED = 3
110	29545	59091	118182	14773 * 32
300	10833	29545	59091	5417 * 32
1200	2708	10833	29545	1354 * 32
2400	1354	2708	10833	667 * 32
4800	677	1354	2708	339 * 32
9600	339	677	1354	169 * 32
19200	169	339	677	36 * 75
38400	85	169	339	52 * 26
57600	56	85	169	32 * 28
115200	28	56	85	16 * 28
230400	14	28	56	8 * 28
460800	7	14	28	4 * 28
921600	*	7	14	2 * 28

Table 5 Divisor needed to generate a given baud rate from 52 MHz based on different HIGHSPEED value

UARTn+0028h SAMPLE_COUNT — UARTn_SAMPLE_COUNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLECOUNT [7:0]							
Type									R/W							
Reset									0							

When HIGHSPEED=3, the sample_count is the threshold value for UART sample counter (sample_num).

Count from 0 to sample_count.

UARTn+002Ch SAMPLE_POINT — UARTn_SAMPLE_POINT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLEPOINT [7:0]							
Type									R/W							
Reset									Ffh							

When HIGHSPEED=3, UART gets the input data when sample_count=sample_num.

e.g. system clock = 13MHz, 921600 = 13000000 / 14
sample_count = 14 and sample point = 7 (sample the central point to decrease the inaccuracy)
The SAMPLE_POINT is usually (SAMPLE_COUNT/2).

UARTn+0034h Rate Fix Address UARTn_RATEFIX_AD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																RXTE_FIX
Type																R/W
Reset																0

rate_fix When you set "rate_fix"(34H[0]), you can transmit and receive data only if the input f16m_en is enable.

UARTn+003Ch Guard time added register UARTn_GUARD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT[3:0]			
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0

GUARD_CNT Guard interval count value. Guard interval = (1/(system clock / **div_step** / div)) * GUARD_CNT.

GUARD_EN Guard interval add enable signal.

0 No guard interval added.

1 Add guard interval after stop bit.

UARTn+0040h Escape character register UARTn_ESCAPE_DAT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT[7:0]							
Type									WO							
Reset									FFh							

ESCAPE_DAT Escape character added before software flow control data and escape character, i.e. if tx data is xon (31h), with esc_en =1, uart transmits data as esc + CEh (~xon).

UARTn+0044h Escape enable register UARTn_ESCAPE_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																R/W
Reset																0

ESC_EN Add escape character in transmitter and remove escape character in receiver by UART.

0 Do not deal with the escape character.

1 Add escape character in transmitter and remove escape character in receiver.

UARTn+0048h Sleep enable register UARTn_SLEEP_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SELLP_EN

Type																R/W
Reset																0

SLEEP_EN For sleep mode issue

- **0** Do not deal with sleep mode indicate signal
- **1** To activate hardware flow control or software control according to software initial setting when chip enters sleep mode. Releasing hardware flow when chip wakes up; but for software control, uart sends xon when awaken and when FIFO does not reach threshold level.

UARTn+004Ch Virtual FIFO enable register — UARTn_VFIFO_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																VFIFO_EN
Type																R/W
Reset																0

VFIFO_EN Virtual FIFO mechanism enable signal.

- **0** Disable VFIFO mode.
- **1** Enable VFIFO mode. When virtual mode is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt for DMA.

UARTn+0050h Rx Trigger Address — UARTn_RXTRI_AD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RXTRIG[3:0]			
Type													R/W			
Reset													0			

RXTRIG **When {rtm,rtl}=2'b11, The Rx FIFO threshold will be Rxtrig.**

UARTn+0054h Fractional Divider LSB Address — UARTn_FRACDIV_L

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									R/W							
Reset									0	0	0	0	0	0	0	0

FRACDIV_L Add sampling count (+1) from state data7 to state data0, in order to contribute fractional divisor.

UARTn+0058h Fractional Divider MSB Address — UARTn_FRACDIV_M

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															FRACDIV_M	
Type															R/W	
Reset															0	0

FRACDIV_M Add sampling count in state stop and state parity, in order to contribute fractional divisor.

FRACDIV_L / FRACDIV_L Add one sampling period to each symbol, in order to increase the baud rate accuracy.

bit_extend register = FRACDIV_L[7:0]
FRACDIV_M[1:0]

Start	d0	d1	d2	d3	d4	d5	d6	d7	Parity	Stop

UARTn+005Ch FIFO Control Register UARTn_FCR_RD

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1	RFTL0	TFTL1	TFTL0	DMA1			FIFOE
Type									RO							RO

Read out UARTn_FCR register.

UARTn+0060h TX Active Enable Address UARTn_TX_ACTIVE_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															TX_PU_EN	TX_OE_EN
Type															R/W	R/W
Reset															0	0

TX_OE_EN Enable UART_TX_OE switching function. TX_OE is to control UART_TX output enable.
TX_PU_EN Enable UART_TX_PU switching function. TX_PU is to control UART_TX pull up enable.

2.14 PCM Controller

2.14.1 Features

- Two clock sources are reserved for PCM circuit. (From internal clock generator, INT_PCM_CLK and EXT_PCM_CLK)
- PCM module can drive a clock out (with fraction-N dividor) to an external codec.
- Up to 4 channels PCM are available. 4 to 128 slots are configurable.
- Each channel supports a-law (8-bit)/u-law (8-bit)/raw-PCM (8-bit and 16-bit) transfer.
- Hardware converter of a-law<->raw-16 and u-law <-> raw-16 are implemented in design.
- Support long (8 cycle)/short (1 cycle)/configurable (intervals are configurable, use to emulate I^2S interface) FSYNC.
- DATA & FSYNC can be driven and sampled by either rising/falling of clock.
- Last bit of DTX can be configured as tri-stated on falling edge.
- Beginning of each slot is configurable by 10-bit registers on each channel.
- 32-byte FIFO are available for each channel
- PCM interface can emulate I2S interface (only 16-bit data-width supported).
- MSB/LSB order is configurable.
- Supports both a-law/u-law (8-bits) →linear PCM(16-bit) and linear PCM(16-bit) → a-law/u-law (8-bit)

2.14.2 Block Diagram

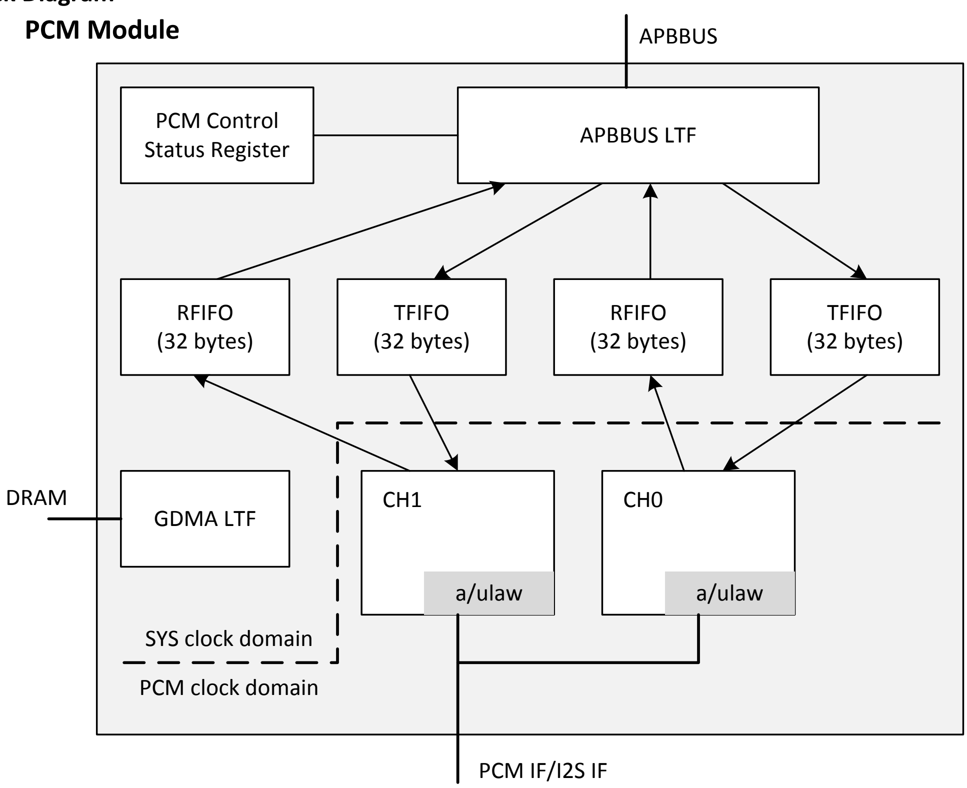


Figure 2-8 PCM Controller Block Diagram

Two clock domains are partitioned in this design. PCM converter (u-law < = > raw-16-bit and A-law < = > raw 16-bit) are implemented in PCM. The threshold of FIFO is configurable. When the threshold is reached, PCM (a) triggers the DMA interface to notify external DMA engine to transfer data, and (b) triggers an interrupt to the host.

The interrupt sources include:

- The threshold is reached.
- FIFO is under-run or over-run.
- A fault is detected at the DMA interface.

The A-law and u-law converter is implemented based on the ITU-G.711 A-law and u-law table. In this design, both A-law/u-law(8-bit) → linear PCM (16-bit) and linear PCM (16-bit) → A-law/u-law (8-bit) are supported.

The data-flow from codec to PCM-controller (Rx-flow) is shown as below:

- The PCM controller latches the data from DRX at the indicated time slot and then writes it to FIFO. If FIFO is full, the data is lost.
- When the Rx-FIFO reaches the threshold, two actions may be taken:
  - When DMA_ENA=1, DMA_REQ is asserted to request a burst transfer. It rechecks the FIFO threshold after DMA_END is asserted by GDMA. (GDMA should be configured before channel is enabled.)
  - Assert the interrupt source to notify the host. The host can check RFIFO_AVAIL information then get back the data from FIFO.

The data flow from the PCM controller to codec (Tx-flow) is shown below. After GDMA is configured, software should configure and enable the PCM channel. The empty FIFO should behave as follows.

- When DMA_ENA=1, DMA_REQ is triggered to request a burst transfer. It then re-checks the FIFO threshold after DMA_END is asserted by GDMA (a burst is completed).
- The Interrupt source is asserted to notify HOST. HOST writes the data to Tx-FIFO. After that, HOST rechecks TFIFO_EMPTY information, and then writes more data if available.

NOTE: When DMA_ENA=1, the burst size of GDMA should be less than the threshold value.

2.14.3 List of Registers

2.14.4 PCM Configuration

PCM Initialization Flow

1. Set PCM_CFG
2. Set CH0/1_CFG
3. Write PCM data to FIFO CH0/1_FIFO
4. Set GLB_CFG to enable the PCM and channel.
5. Set dividor clock
6. Enable clock
7. Monitor FF_STATUS to receive/transmit the other PCM data.

PCM Configuration Examples

Below are some examples of PCM configuration.

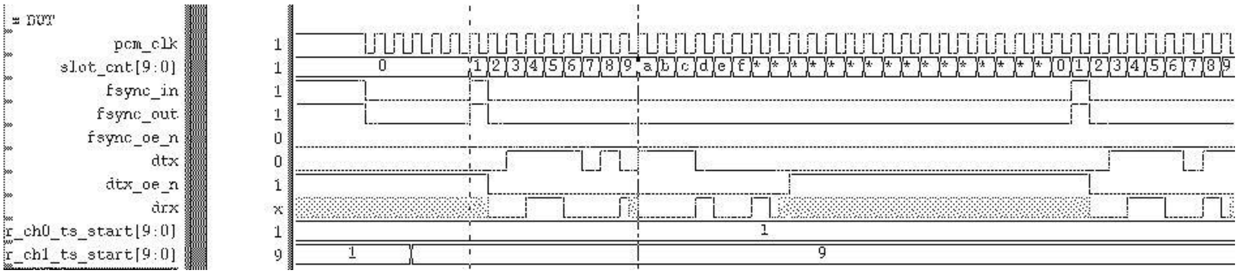
Case 1:

CFG_FSYNC Register: CFG_FSYNC_EN = 0 (PS: fsync is always driven at SLOT_CNT=1)

CH0_CFG Register: TS_START=1

CH1_CFG Register: TS_START=9

PCM_CFG Register: LONG_FSYNC=1’b0, FSYNC_POL=1’b1, DRX_TRI=1’b0, SLOT_MODE=3’b0



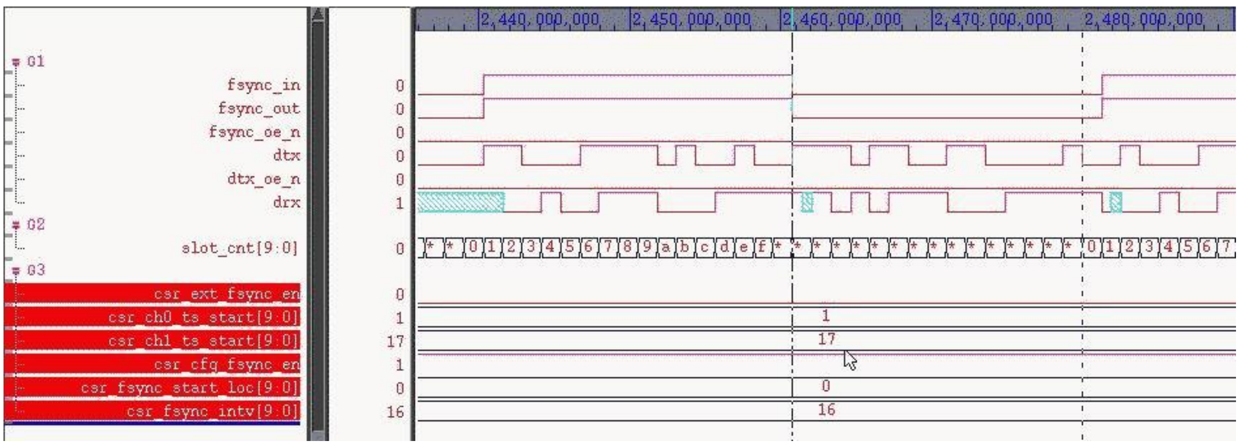
Case 2:

CFG_FSYNC Register: CFG_FSYNC_EN = 1, START_LOC=0, interval=16

CH0_CFG Register: TS_START=1

CH1_CFG Register: TS_START=17

PCM_CFG Register: LONG_FSYNC=1’b0, FSYNC_POL=1’b1, DRX_TRI=1’b0, SLOT_MODE=3’b0, RAW16-bits



Case 3:

CFG_FSYNC Register: CFG_FSYNC_EN = 1, START_LOC=0x1A, interval=2

CH0_CFG Register: TS_START=1 (disable)

CH1_CFG Register: TS_START=0x1A

PCM_CFG Register: LONG_FSYNC=1’b0, FSYNC_POL=1’b0 (LOW active), DRX_TRI=1’b0, SLOT_MODE=3’b0, RAW16-bits



2.14.5 Register

PCM Changes LOG

Revision	Date	Author	Change Log
0.1	2012/10/8	Paddy Wu	Initialization

Module name: PCM Base address: (+10002000h)

Address	Name	Width	Register Function
10002000	GLB_CFG	32	Global Config
10002004	PCM_CFG	32	PCM configuration
10002008	INT_STATUS	32	Interrupt status
1000200C	INT_EN	32	Interrupt enable
10002010	CHA0_FF_STATUS	32	Channel A0(represents channel 0) FIFO status
10002014	CHB0_FF_STATUS	32	Channel B0(represents channel 1) FIFO status
10002020	CHA0_CFG	32	Channel A0(represents channel 0) Config
10002024	CHB0_CFG	32	Channel B0(represents channel 1) Config
10002030	FSYNC_CFG	32	FSYNC config
10002034	CHA0_CFG2	32	Channel A0(represents channel 0) Config
10002038	CHB0_CFG2	32	Channel B0(represents channel 1) Config
10002040	IP_INFO	32	IP version info
10002044	RSV_REG16	32	SPARE REG 16 bits
10002050	DIVCOMP_CFG	32	Dividor Compensation part config
10002054	DIVINT_CFG	32	Dividor Integer part config
10002060	DIGDELAY_CFG	32	Digital delay config
10002080	CH0_FIFO	32	Channel 0 FIFO access point
10002084	CH1_FIFO	32	Channel 1 FIFO access point
10002088	CH2_FIFO	32	Channel 2 FIFO access point
1000208C	CH3_FIFO	32	Channel 3 FIFO access point
10002110	CHA1_FF_STATUS	32	Channel A1(represents channel 3) FIFO status
10002114	CHB1_FF_STATUS	32	Channel B1(represents channel 4) FIFO status
10002120	CHA1_CFG	32	Channel A1(represents channel 3) Config
10002124	CHB1_CFG	32	Channel B1(represents channel 1) Config
10002134	CHA1_CFG2	32	Channel A1(represents channel 3) Config
10002138	CHB1_CFG2	32	Channel B1(represents channel 4) Config

10002000 GLB_CFG Global Config 00440000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCM_EN	DMA_EN	LBK_EN	EXT_LBK_EN	RSV0					RFF_THRES			RSV1	TFF_THRES		

Type	RW	RW	RW	RW	RO					RW			RO	RW		
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV2**												**CH_EN**			
Type	RO												RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	PCM_EN	**PCM Enable** When disabled, all FSM of PCM are cleared to their default value. 0: disable 1: enable
30	DMA_EN	**DMA Enable** 0: Disable the DMA interface, transfer data using software. 1: Enable the DMA interface, transfer data using DMA. 0: disable 1: enable
29	LBK_EN	**loopback enable, loopback path is shown as (Asyn-TXFIFO ->DTX -> DRX->Asyn-RXFIFO)** 0: disable 1: enable
28	EXT_LBK_EN	**loopback enable, loopback path is shown as (Ext-Codec->DRX->DTX->Ext-Codec)** 0: disable 1: enable
27:23	RSV0	**Reserved**
22:20	RFF_THRES	**RXFIFO Threshold** When the threshold is reached, the host/DMA is notified to fill FIFO. The threshold should be >2 and <6. When data in FIFO is under the threshold, the following interrupts and GDMA are triggered. CH0T_THRES, CH0R_THRES, CH1T_THRES, CH1R_THRES (unit: word)
19	RSV1	**Reserved**
18:16	TFF_THRES	**TXFIFO Threshold** When the threshold is reached, the host/DMA is notified to fill FIFO. It should be >2 and <6. When data in FIFO is over the threshold, an interrupt and DMA are triggered. (unit: word)
15:4	RSV2	**Reserved**
3:0	CH_EN	**Channels 3 to 0 Tx and Rx Enable** 0: disable 1: enable

10002004 **PCM_CFG** **PCM configuration** **03000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RSV0**	**CLKOUT_EN**	**RSV1**		**EXT_FSYNC**	**LONG_SYNC**	**FSYNC_POL**	**DTX_TRI**	**RSV2[20:13]**							
Type	RO	RW	RO		RW	RW	RW	RW	RO							
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV2[12:0]**													**SLOT_MODE**		
Type	RO													RW		

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	RSV0	**Reserved**
30	CLKOUT_EN	**PCM Clock Out Enable** 0: A PCM clock is provided from the external Codec/OSC. 1: A PCM clock is provided from the internal dividor. NOTE: Normally, the register should be asserted to 1. Also, it should be asserted after configuring the divider and enabling the divider clock. 0: EXT_CLK 1: INT_DIV
29:28	RSV1	**Reserved**
27	EXT_FSYNC	**FSYNC is provided externally** 0: FSYNC is generated by internal circuit. 1: FSYNC is provided externally
26	LONG_SYNC	**FSYNC Mode** 0: Short FSYNC 1: Long FSYNC
25	FSYNC_POL	**FSYNC Polarity** 0: FSYNC is low active 1: FSYNC is high active
24	DTX_TRI	**DTX Tri-State** Tristates DTX when the clock signal on the last bit is has a falling edge. 0: Non- tristate DTX 1: Tristate DTX
23:3	RSV2	**Reserved**
2:0	SLOT_MODE	**Sets the number of slots in each PCM frame.** 0: 4 slots, PCM clock out/in should be 256 KHz. 1: 8 slots, PCM clock out/in should be 512 KHz. 2: 16 slots, PCM clock out/in should be 1.024 MHz. 3: 32 slots, PCM clock out/in should be 2.048 MHz. 4: 64 slots, PCM clock out/in should be 4.096 MHz. 5:128 slots, PCM clock out/in should be 8.192 MHz. Other: Reserved. NOTE: When using the external clock, the frequency clock should be equal to PCM_clock out. Otherwise, the PCM_CLKin should be 8.192 MHz. 0: _4_SLOT 1: _8_SLOT 2: _16_SLOT 3: _32_SLOT 4: _64_SLOT 5: _128_SLOT

10002008 **INT_STATUS** **Interrupt status** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[23:8]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[7:0]								CHT_DMA_FAULT	CHT_OVRUN	CHT_UNRUN	CHT_THRES	CHR_DMA_FAULT	CHR_OVRUN	CHR_UNRUN	CHR_THRES

Type	RO								W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:8	RSV0	**Reserved**
7	CHT_DMA_FAULT	**Channel Tx DMA Fault Interrupt, Asserts when a fault has been detected in a CH-Tx DMA signal.**
6	CHT_OVRUN	**Channel Tx FIFO Overrun Interrupt, Asserts when the CH-Tx FIFO is overrun.**
5	CHT_UNRUN	**Channel Tx FIFO Underrun Interrupt, Asserts when the CH-Tx FIFO is underrun.**
4	CHT_THRES	**Channel Tx Threshold Interrupt, Asserts when the CH-Tx FIFO is lower than the defined threshold.**
3	CHR_DMA_FAULT	**Channel Rx DMA Fault Interrupt, Asserts when a fault is detected in a CH-Rx DMA signal.**
2	CHR_OVRUN	**Channel Rx Overrun Interrupt, Asserts when the CH-Rx FIFO is overrun.**
1	CHR_UNRUN	**Channel Rx Underrun Interrupt, Asserts when the CH-Rx FIFO is underrun.**
0	CHR_THRES	**Channel Rx Threshold Interrupt, Asserts when the CH-Rx FIFO is lower than the defined threshold.**

1000200C **INT_EN** **Interrupt enable** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[23:8]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[7:0]								INT7_EN	INT6_EN	INT5_EN	INT4_EN	INT3_EN	INT2_EN	INT1_EN	INT0_EN
Type	RO								RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:8	RSV0	**Reserved**
7	INT7_EN	**INT_STATUS[7] Enable,Enables the Channel Tx DMA Fault Interrupt. This interrupt asserts when a fault has been detected in a CH-Tx DMA signal.**
6	INT6_EN	**INT_STATUS[6] Enable,Enables the Channel Tx FIFO Overrun Interrupt. This interrupt asserts when the CH-Tx FIFO is overrun.**
5	INT5_EN	**INT_STATUS[5] Enable,Enables the Channel Tx FIFO Underrun Interrupt. This interrupt asserts when the CH-Tx FIFO is underrun.**
4	INT4_EN	**INT_STATUS[4] Enable,Enables the Channel Tx Threshold Interrupt. This interrupt when the CH-Tx FIFO is lower than the defined threshold.**
3	INT3_EN	**INT_STATUS[3] Enable,Enables the Channel Rx DMA Fault Interrupt. This interrupt when a fault is detected in a CH-Rx DMA signal.**
2	INT2_EN	**INT_STATUS[2] Enable,Enables the Channel Rx Overrun Interrupt. This interrupt when the CH-Rx FIFO is overrun.**
1	INT1_EN	**INT_STATUS[1] Enable,Enables the Channel Rx Underrun Interrupt. This interrupt when the CH-Rx FIFO is under-run.**
0	INT0_EN	**INT_STATUS[0] Enable,Enables the Channel Rx Threshold Interrupt.**

Bit(s)	Name	Description
		This interrupt asserts when the CH-Rx FIFO is lower than the defined threshold.

10002010 **CHA0_FF_STATUS** **Channel A0(represents channel 0) FIFO status** **00100008**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0								CHTX_DMA_FAULT	CHTX_OVRUN	CHTX_UNRUN	CHTX_THRES	CHRX_DMA_FAULT	CHRX_OVRUN	CHRX_UNRUN	CHRX_THRES
Type	RO								W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1								CHRFF_AVCNT				CHTFF_EPCNT			
Type	RO								RO				RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:24	RSV0	Reserved
23	CHTX_DMA_FAULT	Tx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel A0 Tx DMA signal.
22	CHTX_OVRUN	Tx Overrun Interrupt,Asserts when the Channel A0 Tx FIFO is overrun.
21	CHTX_UNRUN	Tx FIFO Underrun Interrupt,Asserts when the Channel A0 Tx FIFO is underrun.
20	CHTX_THRES	Tx FIFO Below Threshold Interrupt,Asserts when the Channel A0 FIFO is lower than the defined threshold.
19	CHRX_DMA_FAULT	Rx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel A0 Rx DMA signal.
18	CHRX_OVRUN	Rx FIFO Overrun Interrupt,Asserts when the Channel A0 Rx FIFO is overrun.
17	CHRX_UNRUN	Rx FIFO Underrun Interrupt,Asserts when the Channel A0 Rx FIFO is underrun.
16	CHRX_THRES	Rx FIFO Below Threshold Interrupt,Asserts when the Channel A0 FIFO is lower than the defined threshold.
15:8	RSV1	Reserved
7:4	CHRFF_AVCNT	Channel A0 RXFIFO Available Space Count,Counts the available space for reads in channel A0 RXFIFO.(unit: word)
3:0	CHTFF_EPCNT	Channel A0 TXFIFO Available Space Count,Counts the available space for writes in channel A0 TXFIFO.(unit: word)

10002014 **CHB0_FF_STATUS** **Channel B0(represents channel 1) FIFO status** **00100008**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0								CHTX_DMA_FAULT	CHTX_OVRUN	CHTX_UNRUN	CHTX_THRES	CHRX_DMA_FAULT	CHRX_OVRUN	CHRX_UNRUN	CHRX_THRES

Type	RO								W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV1**								**CHRFF_AVCNT**				**CHTFF_EPCNT**			
Type	RO								RO				RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:24	RSV0	**Reserved**
23	CHTX_DMA_FAULT	**Tx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel B0 Tx DMA signal.**
22	CHTX_OVRUN	**Tx Overrun Interrupt,Asserts when the Channel B0 Tx FIFO is overrun.**
21	CHTX_UNRUN	**Tx FIFO Underrun Interrupt,Asserts when the Channel B0 Tx FIFO is underrun.**
20	CHTX_THRES	**Tx FIFO Below Threshold Interrupt,Asserts when the Channel B0 FIFO is lower than the defined threshold.**
19	CHRX_DMA_FAULT	**Rx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel B0 Rx DMA signal.**
18	CHRX_OVRUN	**Rx FIFO Overrun Interrupt,Asserts when the Channel B0 Rx FIFO is overrun.**
17	CHRX_UNRUN	**Rx FIFO Underrun Interrupt,Asserts when the Channel B0 Rx FIFO is underrun.**
16	CHRX_THRES	**Rx FIFO Below Threshold Interrupt,Asserts when the Channel B0 FIFO is lower than the defined threshold.**
15:8	RSV1	**Reserved**
7:4	CHRFF_AVCNT	**Channel B0 RXFIFO Available Space Count,Counts the available space for reads in channel B0 RXFIFO.(unit: word)**
3:0	CHTFF_EPCNT	**Channel B0 TXFIFO Available Space Count,Counts the available space for writes in channel B0 TXFIFO.(unit: word)**

10002020 CHA0_CFG Channel A0(represents channel 0) Config 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RSV0**		**CMP_MODE**			**RSV1[16:6]**										
Type	RO		RW			RO										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV1[5:0]**						**TS_START**									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:30	RSV0	**Reserved**
29:27	CMP_MODE	**Compression Mode** Sets the conversion method for the hardware converter to compress raw data. 000: Disable HW converter, linear raw data (16-bit) 010: Disable HW converter, linear raw data (8-bit), A-law or u-law (8-bit) 011: Reserved 100: Enable HW converter, raw data(16-bit) U-law mode (8-bit) (PCM bus in compressed format) 101: Enable HW converter, u-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 110: Enable HW converter, raw data (16-bit) A-law mode (8-bit) (PCM bus in

Bit(s)	Name	Description
		compressed format) 111: Enable HW converter, A-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 0: DIS_CONV16 2: DIS_CONV8 4: EN_ULW2R 5: EN_R2ULW 6: EN_ALW2R 7: EN_R2ALW
26:10	RSV1	**Reserved**
9:0	TS_START	**Timeslot starting location** (unit: clock cycles)

10002024 CHB0_CFG Channel B0(represents channel 1) Config 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0		CMP_MODE			RSV1[16:6]										
Type	RO		RW			RO										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1[5:0]						TS_START									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:30	RSV0	**Reserved**
29:27	CMP_MODE	**Compression Mode** Sets the conversion method for the hardware converter to compress raw data. 000: Disable HW converter, linear raw data (16-bit) 010: Disable HW converter, linear raw data (8-bit), A-law or u-law (8-bit) 011: Reserved 100: Enable HW converter, raw data(16-bit) U-law mode (8-bit) (PCM bus in compressed format) 101: Enable HW converter, u-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 110: Enable HW converter, raw data (16-bit) A-law mode (8-bit) (PCM bus in compressed format) 111: Enable HW converter, A-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 0: DIS_CONV16 2: DIS_CONV8 4: EN_ULW2R 5: EN_R2ULW 6: EN_ALW2R 7: EN_R2ALW
26:10	RSV1	**Reserved**
9:0	TS_START	**Timeslot starting location** (unit: clock cycles)

10002030 FSYNC_CFG FSYNC config 28000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	CFG_FSYNC_EN	POS_CAP_DT	POS_DRV_DT	POS_CAP_FSYNC	POS_DRV_FSYNC	RSV0					RSV1[11:6]					
Type	RW	RW	RW	RW	RW	RO					RO					
Reset	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1[5:0]						FSYNC_INTV									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	CFG_FSYNC_EN	**Enables configurable FSYNC.**
30	POS_CAP_DT	**Positive Edge Capture Data, Sets the PCM controller to capture data on the negative or positive edge of the PCM clock. NOTE: This configuration should be 0 if DTX_TRI=1.**
29	POS_DRV_DT	**Positive Edge Drive Data, Sets the PCM controller to drive data on the negative or positive edge of the PCM clock.**
28	POS_CAP_FSYNC	**Positive Edge Capture FSYNC, Sets the PCM controller to capture FSYNC on the positive or negative edge of the PCM clock.**
27	POS_DRV_FSYNC	**Positive Edge Driver FSYNC, Sets the PCM controller to drive FSYNC on the negative or positive edge of the PCM clock.**
26:22	RSV0	**Reserved**
21:10	RSV1	**Reserved**
9:0	FSYNC_INTV	**Interval when FSYNC may be configured.** (unit: clock cycles)

10002034 CHA0_CFG2 Channel A0(represents channel 0) Config 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[11:0]												CH_RXFF_CLR	CH_TXFF_CLR	RSV1	CH_LSB
Type	RO												RW	RW	RO	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RSV0	**Reserved**
3	CH_RXFF_CLR	**Channel A0 Rx FIFO Clear** 0: Normal operation 1: Clear this bit
2	CH_TXFF_CLR	**Channel A0 Tx FIFO Clear** 0: Normal operation 1: Clear this bit
1	RSV1	**Reserved**
0	CH_LSB	**Enable CH A0 Tx in LSB order.**

10002038 **CHB0_CFG2** **Channel B0(represents channel 1) Config** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[11:0]												CH_RXFF_CLR	CH_TXFF_CLR	RSV1	CH_LSB
Type	RO												RW	RW	RO	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RSV0	**Reserved**
3	CH_RXFF_CLR	**Channel B0 Rx FIFO Clear** 0: Normal operation 1: Clear this bit
2	CH_TXFF_CLR	**Channel B0 Tx FIFO Clear** 0: Normal operation 1: Clear this bit
1	RSV1	**Reserved**
0	CH_LSB	**Enable CH B0 Tx in LSB order.**

10002040 **IP_INFO** **IP version info** **00000401**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MAX_CH								VER							
Type	RO								RO							
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RSV0	**Reserved**
15:8	MAX_CH	**Maximum channel number.**
7:0	VER	**Version of this PCM Controller**

10002044 **RSV_REG16** **SPARE REG 16 bits** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPARE_REG															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RSV0	**Reserved**
15:0	SPARE_REG	**Spare register for future use**

10002050 **DIVCOMP_CFG** **Dividor Compensation part config** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CLK_EN	RSV0[22:8]														
Type	RW	RO														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSV0[7:0]								DIVCOMP							
Type	RO								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	CLK_EN	**Clock Enable** Enables setting of the PCM interface clock based on DIVCOMP and DIVINT parameters.
30:8	RSV0	**Reserved**
7:0	DIVCOMP	**A parameter in an equation which determines FREQOUT. See DIVINT.**

10002054 **DIVINT_CFG** **Dividor Integer part config** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[21:6]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSV0[5:0]						DIVINT									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:10	RSV0	**Reserved**
9:0	DIVINT	**A parameter in an equation which determines FREQOUT.** Formula: FREQOUT = 1/(FREQIN*2*(DIVINT+DIVCOMP /(2^8))) FREQIN is always fixed to 40 MHz.

10002060 **DIGDELAY_CFG** **Digital delay config** **00000002**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TXD_CLR_GL	CHEN_CLR_GL	RSV0			TXD_GLT_ST	RSV1			CHENN_GLT_S	RSV2			CHENP_GLT_S	RSV3	CHENPD_GLT_S

	T	T								T				T		T
Type	RW	RW	RO			RW	RO			RW	RO			RW	RO	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TXD_DIGDLY_EN**	**RSV4**		**TXD_DLYVAL**					**CHEN_DIGDLY_EN**	**RSV5**		**CHEN_DLYVAL**				
Type	RW	RO		RW					RW	RO		RW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit(s)	Name	Description
31	TXD_CLR_GLT	**TXD Clear Glitch Flag** Clears the glitch detected flag for TXD. 0: No effect. 1: Clear the flag.
30	CHEN_CLR_GLT	**Channel Enable (CHEN) Clear Glitch Flag** Clears the glitch detected flag for CHEN. 0: No effect . 1: Clear the flag.
29:27	RSV0	**Reserved**
26	TXD_GLT_ST	**TXD Glitch Status** Indicates if a glitch is detected in a TXD signal. It can be cleared by bit[31]. 0: Not detected. 1: Detected
25:23	RSV1	**Reserved**
22	CHENN_GLT_ST	**CHEN Negative Glitch Status** Indicates if a glitch is detected in a CHEN signal. It can be cleared by bit[30] (negedge sample). 0: Not detected. 1: Detected
21:19	RSV2	**Reserved**
18	CHENP_GLT_ST	**CHEN Positive Glitch Status** Indicates if a glitch is detected in a CHEN signal. It can be cleared by bit[30] (posedge sample). 0: Not detected. 1: Detected
17	RSV3	**Reserved**
16	CHENPD_GLT_ST	**CHEN Positive Delay Glitch Status** Indicates if a glitch is detected in a CHEN signal. It can be cleared by bit[30] (posedge sample, delay 1 cycle). 0: Not detected. 1: Detected
15	TXD_DIGDLY_EN	**TXD Digital Delay Enable** Enables digital delay path. 0: Disable 1: Enable
14:13	RSV4	**Reserved**
12:8	TXD_DLYVAL	**Delay Count Value** The description is the same as the CHEN_DLYVAL field in this register. CHEN Digital Delay Enable, Enables the digital delay path. 0: Disable 1: Enable
7	CHEN_DIGDLY_EN	**CHEN Digital Delay Enable** Enables the digital delay path.

Bit(s)	Name	Description
		0: Disable 1: Enable
6:5	RSV5	**Reserved**
4:0	CHEN_DLYVAL	**Delay Count Value** The delay error = CLK_PERIOD * (SYNC_DELAY + SYNC_DELTA + (DLYCNT_CFG) + 1) For example, DLYCNT_CFG = 4, (SYNC_DELAY is always fixed to 4) Final Delay = CLK_PERIOD * (2 + (-1/0/+1) + (4) + 1) = CLK_PERIOD * (6/7/8)= CLK_PERIOD * (6 to 8) = 25 ns to 33.3 ns NOTE: Period is 1/240 MHz = 4.1667 ns in MT7620.

10002080 CH0_FIFO Channel 0 FIFO access point 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CH0_FIFO[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CH0_FIFO[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CH0_FIFO	**Channel 0 FIFO access point**

10002084 CH1_FIFO Channel 1 FIFO access point 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CH1_FIFO[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CH1_FIFO[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CH1_FIFO	**Channel 1 FIFO access point**

10002088 CH2_FIFO Channel 2 FIFO access point 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CH2_FIFO[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	CH2_FIFO[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CH2_FIFO	**Channel 2 FIFO access point**

1000208C CH3_FIFO Channel 3 FIFO access point 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CH3_FIFO[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CH3_FIFO[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CH3_FIFO	**Channel 3 FIFO access point**

10002110 CHA1_FF_STATUS Channel A1(represents channel 3) FIFO status 00100008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0								CHTX_DMA_FAULT	CHTX_OVRUN	CHTX_UNRUN	CHTX_THRES	CHRX_DMA_FAULT	CHRX_OVRUN	CHRX_UNRUN	CHRX_THRES
Type	RO								W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1								CHRFF_AVCNT				CHTFF_EPCNT			
Type	RO								RO				RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:24	RSV0	**Reserved**
23	CHTX_DMA_FAULT	**Tx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel A1 Tx DMA signal.**
22	CHTX_OVRUN	**Tx Overrun Interrupt,Asserts when the Channel A0 Tx FIFO is overrun.**
21	CHTX_UNRUN	**Tx FIFO Underrun Interrupt,Asserts when the Channel A1 Tx FIFO is underrun.**
20	CHTX_THRES	**Tx FIFO Below Threshold Interrupt,Asserts when the Channel A0 FIFO is lower than the defined threshold.**
19	CHRX_DMA_FAULT	**Rx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel A1 Rx DMA signal.**
18	CHRX_OVRUN	**Rx FIFO Overrun Interrupt,Asserts when the Channel A1 Rx FIFO is overrun.**
17	CHRX_UNRUN	**Rx FIFO Underrun Interrupt,Asserts when the Channel A1 Rx FIFO is underrun.**

Bit(s)	Name	Description
16	CHRX_THRES	**Rx FIFO Below Threshold Interrupt,Asserts when the Channel A1 FIFO is lower than the defined threshold.**
15:8	RSV1	**Reserved**
7:4	CHRFF_AVCNT	**Channel A1 RXFIFO Available Space Count,Counts the available space for reads in channel A1 RXFIFO.(unit: word)**
3:0	CHTFF_EPCNT	**Channel A1 TXFIFO Available Space Count,Counts the available space for writes in channel A1 TXFIFO.(unit: word)**

10002114	CHB1_FF_STATUS	Channel B1(represents channel 4) FIFO status	00100008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0								CHTX_DMA_FAULT	CHTX_OVRUN	CHTX_UNRUN	CHTX_THRES	CHRX_DMA_FAULT	CHRX_OVRUN	CHRX_UNRUN	CHRX_THRES
Type	RO								W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1								CHRFF_AVCNT				CHTFF_EPCNT			
Type	RO								RO				RO			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit(s)	Name	Description
31:24	RSV0	**Reserved**
23	CHTX_DMA_FAULT	**Tx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel B1 Tx DMA signal.**
22	CHTX_OVRUN	**Tx Overrun Interrupt,Asserts when the Channel B0 Tx FIFO is overrun.**
21	CHTX_UNRUN	**Tx FIFO Underrun Interrupt,Asserts when the Channel B1 Tx FIFO is underrun.**
20	CHTX_THRES	**Tx FIFO Below Threshold Interrupt,Asserts when the Channel B1 FIFO is lower than the defined threshold.**
19	CHRX_DMA_FAULT	**Rx DMA Fault Detected Interrupt,Asserts when a fault is detected in a Channel B1 Rx DMA signal.**
18	CHRX_OVRUN	**Rx FIFO Overrun Interrupt,Asserts when the Channel B1 Rx FIFO is overrun.**
17	CHRX_UNRUN	**Rx FIFO Underrun Interrupt,Asserts when the Channel B1 Rx FIFO is underrun.**
16	CHRX_THRES	**Rx FIFO Below Threshold Interrupt,Asserts when the Channel B1 FIFO is lower than the defined threshold.**
15:8	RSV1	**Reserved**
7:4	CHRFF_AVCNT	**Channel B1 RXFIFO Available Space Count,Counts the available space for reads in channel B1 RXFIFO.(unit: word)**
3:0	CHTFF_EPCNT	**Channel B1 TXFIFO Available Space Count,Counts the available space for writes in channel B1 TXFIFO.(unit: word)**

10002120	CHA1_CFG	Channel A1(represents channel 3) Config	00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RSV0		CMP_MODE			RSV1[16:6]										
Type	RO		RW			RO										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1[5:0]						TS_START									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:30	RSV0	**Reserved**
29:27	CMP_MODE	**Compression Mode** Sets the conversion method for the hardware converter to compress raw data. 000: Disable HW converter, linear raw data (16-bit) 010: Disable HW converter, linear raw data (8-bit), A-law or u-law (8-bit) 011: Reserved 100: Enable HW converter, raw data(16-bit) U-law mode (8-bit) (PCM bus in compressed format) 101: Enable HW converter, u-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 110: Enable HW converter, raw data (16-bit) A-law mode (8-bit) (PCM bus in compressed format) 111: Enable HW converter, A-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 0: DIS_CONV16 2: DIS_CONV8 4: EN_ULW2R 5: EN_R2ULW 6: EN_ALW2R 7: EN_R2ALW
26:10	RSV1	**Reserved**
9:0	TS_START	**Timeslot starting location** (unit: clock cycles)

10002124 CHB1_CFG Channel B1(represents channel 1) Config 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0		CMP_MODE			RSV1[16:6]										
Type	RO		RW			RO										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV1[5:0]						TS_START									
Type	RO						RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:30	RSV0	**Reserved**
29:27	CMP_MODE	**Compression Mode** Sets the conversion method for the hardware converter to compress raw data. 000: Disable HW converter, linear raw data (16-bit) 010: Disable HW converter, linear raw data (8-bit), A-law or u-law (8-bit) 011: Reserved 100: Enable HW converter, raw data(16-bit) U-law mode (8-bit) (PCM bus in compressed format) 101: Enable HW converter, u-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format)

Bit(s)	Name	Description
		110: Enable HW converter, raw data (16-bit) A-law mode (8-bit) (PCM bus in compressed format) 111: Enable HW converter, A-law mode (8-bit) raw data (16-bit) (PCM bus in raw, 16-bit format) 0: DIS_CONV16 2: DIS_CONV8 4: EN_ULW2R 5: EN_R2ULW 6: EN_ALW2R 7: EN_R2ALW
26:10	RSV1	**Reserved**
9:0	TS_START	**Timeslot starting location** (unit: clock cycles)

10002134 CHA1_CFG2 Channel A1(represents channel 3) Config 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[11:0]												CH_RXFF_CLR	CH_TXFF_CLR	RSV1	CH_LSB
Type	RO												RW	RW	RO	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RSV0	**Reserved**
3	CH_RXFF_CLR	**Channel A1 Rx FIFO Clear** 0: Normal operation 1: Clear this bit
2	CH_TXFF_CLR	**Channel A1 Tx FIFO Clear** 0: Normal operation 1: Clear this bit
1	RSV1	**Reserved**
0	CH_LSB	**Enable CH A1 Tx in LSB order.**

10002138 CHB1_CFG2 Channel B1(represents channel 4) Config 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RSV0[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RSV0[11:0]												CH_RXFF_CLR	CH_TXFF_CLR	RSV1	CH_LSB
Type	RO												RW	RW	RO	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RSV0	**Reserved**
3	CH_RXFF_CLR	**Channel B1 Rx FIFO Clear** 0: Normal operation 1: Clear this bit
2	CH_TXFF_CLR	**Channel B1 Tx FIFO Clear** 0: Normal operation 1: Clear this bit
1	RSV1	**Reserved**
0	CH_LSB	**Enable CH B1 Tx in LSB order.**

2.15 Generic DMA Controller

2.15.1 Features

- Supports 16 DMA channels
- Supports 32 bit address.
- Maximum 65535 byte transfer
- Programmable DMA burst size (1, 2, 4, 8, 16 double word burst)
- Supports memory to memory, memory to peripheral, peripheral to memory, peripheral to peripheral transfers.
- Supports continuous mode.
- Supports division of target transfer count into 1 to 256 segments
- Support for combining different channels into a chain.
- Programmable hardware channel priority.
- Interrupts for each channel.

2.15.2 Block Diagram

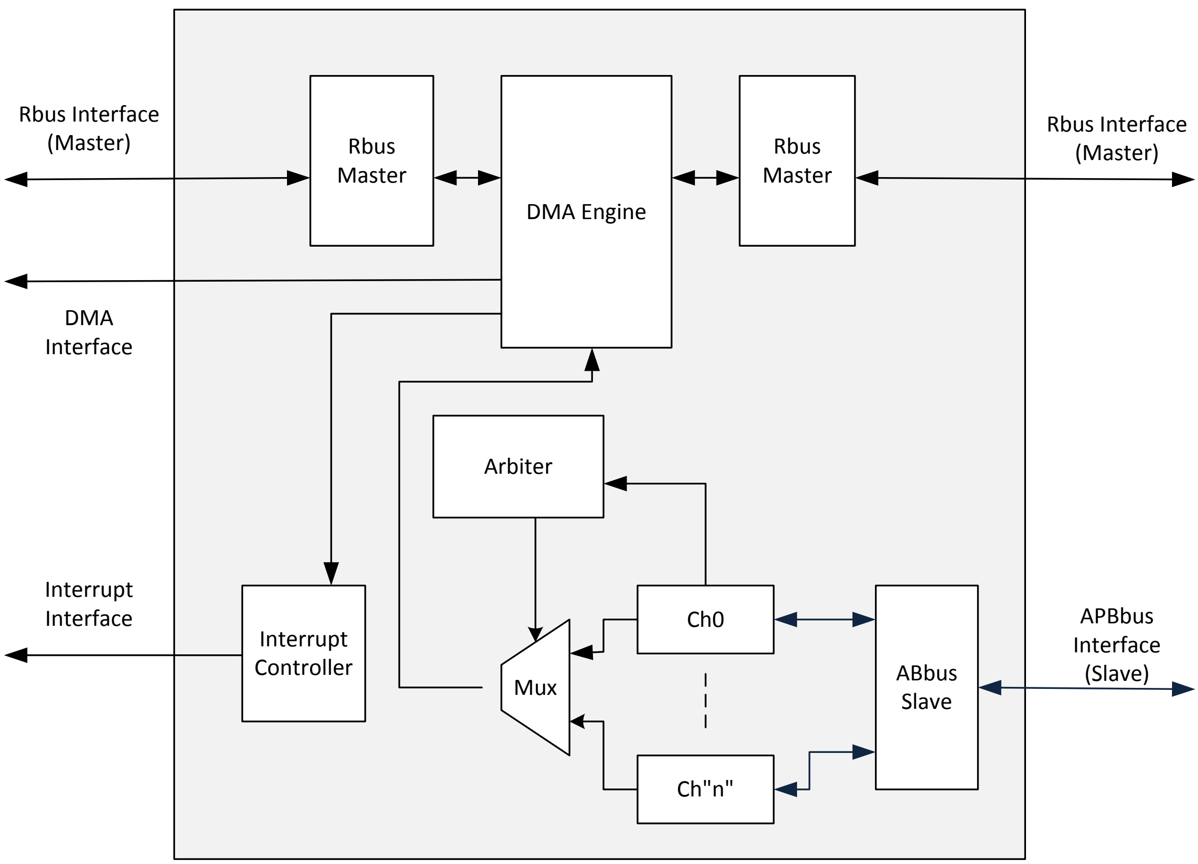


Figure 2-9 Generic DMA Controller Block Diagram

2.15.3 Peripheral Channel Connection

Channel number	Peripheral
0	Reserved
1	Reserved

Channel number	Peripheral
2	I2S Controller (TXDMA)
3	I2S Controller (RXDMA)
4	PCM Controller (RDMA, channel-0)
5	PCM Controller (RDMA, channel-1)
6	PCM Controller (TDMA, channel-0)
7	PCM Controller (TDMA, channel-1)
8	PCM Controller (RDMA, channel-2)
9	PCM Controller (RDMA, channel-3)
10	PCM Controller (TDMA, channel-2)
11	PCM Controller (TDMA, channel-3)
12	SPI Controller (RXDMA)
13	SPI Controller (TXDMA)
8 to 15	Reserved

2.15.4 Registers

GDMA Changes LOG

Revision	Date	Author	Change Log
0.1	2012/10/15	Mark Wang	Initialization

Module name: GDMA Base address: (+10002800h)

Address	Name	Width	Register Function
10002800	GDMA_SA_0	32	Source Address of GDMA Channel 0
10002804	GDMA_DA_0	32	Destination Address of GDMA Channel 0
10002808	GDMA_CT0_0	32	Control Register 0 of GDMA Channel 0
1000280C	GDMA_CT1_0	32	Control Register 1 of GDMA Channel 0
10002810	GDMA_SA_1	32	Source Address of GDMA Channel 1
10002814	GDMA_DA_1	32	Destination Address of GDMA Channel 1
10002818	GDMA_CT0_1	32	Control Register 0 of GDMA Channel 1
1000281C	GDMA_CT1_1	32	Control Register 1 of GDMA Channel 1
10002820	GDMA_SA_2	32	Source Address of GDMA Channel 2
10002824	GDMA_DA_2	32	Destination Address of GDMA Channel 2
10002828	GDMA_CT0_2	32	Control Register 0 of GDMA Channel 2
1000282C	GDMA_CT1_2	32	Control Register 1 of GDMA Channel 2
10002830	GDMA_SA_3	32	Source Address of GDMA Channel 3
10002834	GDMA_DA_3	32	Destination Address of GDMA Channel 3
10002838	GDMA_CT0_3	32	Control Register 0 of GDMA Channel 3
1000283C	GDMA_CT1_3	32	Control Register 1 of GDMA Channel 3
10002840	GDMA_SA_4	32	Source Address of GDMA Channel 4
10002844	GDMA_DA_4	32	Destination Address of GDMA Channel 4
10002848	GDMA_CT0_4	32	Control Register 0 of GDMA Channel 4
1000284C	GDMA_CT1_4	32	Control Register 1 of GDMA Channel 4
10002850	GDMA_SA_5	32	Source Address of GDMA Channel 5

10002854	GDMA_DA_5	32	Destination Address of GDMA Channel 5
10002858	GDMA_CT0_5	32	Control Register 0 of GDMA Channel 5
1000285C	GDMA_CT1_5	32	Control Register 1 of GDMA Channel 5
10002860	GDMA_SA_6	32	Source Address of GDMA Channel 6
10002864	GDMA_DA_6	32	Destination Address of GDMA Channel 6
10002868	GDMA_CT0_6	32	Control Register 0 of GDMA Channel 6
1000286C	GDMA_CT1_6	32	Control Register 1 of GDMA Channel 6
10002870	GDMA_SA_7	32	Source Address of GDMA Channel 7
10002874	GDMA_DA_7	32	Destination Address of GDMA Channel 7
10002878	GDMA_CT0_7	32	Control Register 0 of GDMA Channel 7
1000287C	GDMA_CT1_7	32	Control Register 1 of GDMA Channel 7
10002880	GDMA_SA_8	32	Source Address of GDMA Channel 8
10002884	GDMA_DA_8	32	Destination Address of GDMA Channel 8
10002888	GDMA_CT0_8	32	Control Register 0 of GDMA Channel 8
1000288C	GDMA_CT1_8	32	Control Register 1 of GDMA Channel 8
10002890	GDMA_SA_9	32	Source Address of GDMA Channel 9
10002894	GDMA_DA_9	32	Destination Address of GDMA Channel 9
10002898	GDMA_CT0_9	32	Control Register 0 of GDMA Channel 9
1000289C	GDMA_CT1_9	32	Control Register 1 of GDMA Channel 9
100028A0	GDMA_SA_10	32	Source Address of GDMA Channel 10
100028A4	GDMA_DA_10	32	Destination Address of GDMA Channel 10
100028A8	GDMA_CT0_10	32	Control Register 0 of GDMA Channel 10
100028AC	GDMA_CT1_10	32	Control Register 1 of GDMA Channel 10
100028B0	GDMA_SA_11	32	Source Address of GDMA Channel 11
100028B4	GDMA_DA_11	32	Destination Address of GDMA Channel 11
100028B8	GDMA_CT0_11	32	Control Register 0 of GDMA Channel 11
100028BC	GDMA_CT1_11	32	Control Register 1 of GDMA Channel 11
100028C0	GDMA_SA_12	32	Source Address of GDMA Channel 12
100028C4	GDMA_DA_12	32	Destination Address of GDMA Channel 12
100028C8	GDMA_CT0_12	32	Control Register 0 of GDMA Channel 12
100028CC	GDMA_CT1_12	32	Control Register 1 of GDMA Channel 12
100028D0	GDMA_SA_13	32	Source Address of GDMA Channel 13
100028D4	GDMA_DA_13	32	Destination Address of GDMA Channel 13
100028D8	GDMA_CT0_13	32	Control Register 0 of GDMA Channel 13
100028DC	GDMA_CT1_13	32	Control Register 1 of GDMA Channel 13
100028E0	GDMA_SA_14	32	Source Address of GDMA Channel 14
100028E4	GDMA_DA_14	32	Destination Address of GDMA Channel 14
100028E8	GDMA_CT0_14	32	Control Register 0 of GDMA Channel 14
100028EC	GDMA_CT1_14	32	Control Register 1 of GDMA Channel 14
100028F0	GDMA_SA_15	32	Source Address of GDMA Channel 15
100028F4	GDMA_DA_15	32	Destination Address of GDMA Channel 15
100028F8	GDMA_CT0_15	32	Control Register 0 of GDMA Channel 15
100028FC	GDMA_CT1_15	32	Control Register 1 of GDMA Channel 15
10002A00	GDMA_UNMASK_INTSTS	32	Unmask Fail Interrupt Status
10002A04	GDMA_DONE_INTSTS	32	Segment Done Interrupt Status
10002A20	GDMA_GCT	32	Global Control

10002A30	GDMA_PERI_ADDR_START_0	32	**Peripheral Region 0 Starting Address**
10002A34	GDMA_PERI_ADDR_END_0	32	**Peripheral Region 0 End Address**
10002A38	GDMA_PERI_ADDR_START_1	32	**Peripheral Region 1 Starting Address**
10002A3C	GDMA_PERI_ADDR_END_1	32	**Peripheral Region 1 End Address**
10002A40	GDMA_PERI_ADDR_START_2	32	**Peripheral Region 2 Starting Address**
10002A44	GDMA_PERI_ADDR_END_2	32	**Peripheral Region 2 End Address**
10002A48	GDMA_PERI_ADDR_START_3	32	**Peripheral Region 3 Starting Address**
10002A4C	GDMA_PERI_ADDR_END_3	32	**Peripheral Region 3 End Address**

10002800 GDMA_SA_0 Source Address of GDMA Channel 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002804 GDMA_DA_0 Destination Address of GDMA Channel 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002808 GDMA_CT0_0 Control Register 0 of GDMA Channel 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000280C **GDMA_CT1_0** **Control Register 1 of GDMA Channel 0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT	CH_UNMASK_F	CH_MASK

														_EN	AIL_IN T_E N	
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002810 **GDMA_SA_1** **Source Address of GDMA Channel 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SOURCE_ADDR[31:16]**															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002814 GDMA_DA_1 Destination Address of GDMA Channel 1 **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002818 GDMA_CT0_1 Control Register 0 of GDMA Channel 1 **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs

Bit(s)	Name	Description
		2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000281C **GDMA_CT1_1** **Control Register 1 of GDMA Channel 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0

Bit(s)	Name	Description
		1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002820 **GDMA_SA_2** **Source Address of GDMA Channel 2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002824 **GDMA_DA_2** **Destination Address of GDMA Channel 2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002828 GDMA_CT0_2 Control Register 0 of GDMA Channel 2 **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000282C **GDMA_CT1_2** **Control Register 1 of GDMA Channel 2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this**

Bit(s)	Name	Description
		field is clear by HW/SW. 0: Channel is not masked 1: Channel is masked

10002830 **GDMA_SA_3** **Source Address of GDMA Channel 3** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SOURCE_ADDR[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SOURCE_ADDR[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002834 **GDMA_DA_3** **Destination Address of GDMA Channel 3** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DEST_ADDR[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DEST_ADDR[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002838 **GDMA_CT0_3** **Control Register 0 of GDMA Channel 3** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TARGET_BYTE_CNT**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CURR_SEGMENT**								**SOURCE_ADDR_MODE**	**DEST_ADDR_MODE**	**BURST_SIZE**			**SEGMENT_DONE_INT_EN**	**CH_EN**	**SW_MODE_EN**
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000283C **GDMA_CT1_3** **Control Register 1 of GDMA Channel 3** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a**

Bit(s)	Name	Description
		multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002840 **GDMA_SA_4** **Source Address of GDMA Channel 4** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002844 **GDMA_DA_4** **Destination Address of GDMA Channel 4** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002848 **GDMA_CT0_4** **Control Register 0 of GDMA Channel 4** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable

Bit(s)	Name	Description
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000284C **GDMA_CT1_4** **Control Register 1 of GDMA Channel 4** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1

Bit(s)	Name	Description
		n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002850 GDMA_SA_5 Source Address of GDMA Channel 5 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002854 GDMA_DA_5 Destination Address of GDMA Channel 5 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002858 GDMA_CT0_5 Control Register 0 of GDMA Channel 5 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000285C GDMA_CT1_5 Control Register 1 of GDMA Channel 5 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_	CH_UNMAS	CH_MASK

		EN												INT_EN	K_FAIL_INT_EN	
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002860 **GDMA_SA_6** **Source Address of GDMA Channel 6** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SOURCE_ADDR[31:16]**															

Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002864 GDMA_DA_6 Destination Address of GDMA Channel 6 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002868 GDMA_CT0_6 Control Register 0 of GDMA Channel 6 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW

Bit(s)	Name	Description
		1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000286C GDMA_CT1_6 Control Register 1 of GDMA Channel 6 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request**

Bit(s)	Name	Description
		0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002870 GDMA_SA_7 Source Address of GDMA Channel 7 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002874 GDMA_DA_7 Destination Address of GDMA Channel 7 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002878 **GDMA_CT0_7** **Control Register 0 of GDMA Channel 7** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000287C GDMA_CT1_7 Control Register 1 of GDMA Channel 7 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this**

Bit(s)	Name	Description
		field is clear by HW/SW. 0: Channel is not masked 1: Channel is masked

10002880 **GDMA_SA_8** **Source Address of GDMA Channel 8** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002884 **GDMA_DA_8** **Destination Address of GDMA Channel 8** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002888 **GDMA_CT0_8** **Control Register 0 of GDMA Channel 8** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000288C GDMA_CT1_8 Control Register 1 of GDMA Channel 8 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a**

Bit(s)	Name	Description
		multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002890 GDMA_SA_9 Source Address of GDMA Channel 9 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

10002894 GDMA_DA_9 Destination Address of GDMA Channel 9 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

10002898 GDMA_CT0_9 Control Register 0 of GDMA Channel 9 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable

Bit(s)	Name	Description
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

1000289C GDMA_CT1_9 Control Register 1 of GDMA Channel 9 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1

Bit(s)	Name	Description
		n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

100028A0 **GDMA_SA_10** **Source Address of GDMA Channel 10** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

100028A4 **GDMA_DA_10** **Destination Address of GDMA Channel 10** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

100028A8 **GDMA_CT0_10** **Control Register 0 of GDMA Channel 10** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

100028AC **GDMA_CT1_10** **Control Register 1 of GDMA Channel 10** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_	CH_UNMAS	CH_MASK

		EN												INT_EN	K_FAIL_INT_EN	
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

100028B0 **GDMA_SA_11** **Source Address of GDMA Channel 11** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SOURCE_ADDR[31:16]**															

Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SOURCE_ADDR[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

100028B4 GDMA_DA_11 Destination Address of GDMA Channel 11 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DEST_ADDR[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DEST_ADDR[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

100028B8 GDMA_CT0_11 Control Register 0 of GDMA Channel 11 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TARGET_BYTE_CNT**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CURR_SEGMENT**								**SOURCE_ADDR_MODE**	**DEST_ADDR_MODE**	**BURST_SIZE**			**SEGMENT_DONE_INT_EN**	**CH_EN**	**SW_MODE_EN**
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW

Bit(s)	Name	Description
		1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

100028BC **GDMA_CT1_11** **Control Register 1 of GDMA Channel 11** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request**

Bit(s)	Name	Description
		0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

100028C0 **GDMA_SA_12** **Source Address of GDMA Channel 12** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

100028C4 **GDMA_DA_12** **Destination Address of GDMA Channel 12** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

100028C8 **GDMA_CT0_12** **Control Register 0 of GDMA Channel 12** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

100028CC	GDMA_CT1_12	Control Register 1 of GDMA Channel 12	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this**

Bit(s)	Name	Description
		field is clear by HW/SW. 0: Channel is not masked 1: Channel is masked

100028D0 **GDMA_SA_13** **Source Address of GDMA Channel 13** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

100028D4 **GDMA_DA_13** **Destination Address of GDMA Channel 13** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

100028D8 **GDMA_CT0_13** **Control Register 0 of GDMA Channel 13** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

100028DC **GDMA_CT1_13** **Control Register 1 of GDMA Channel 13** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a**

Bit(s)	Name	Description
		multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

100028E0 GDMA_SA_14 Source Address of GDMA Channel 14 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

100028E4 **GDMA_DA_14** **Destination Address of GDMA Channel 14** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

100028E8 **GDMA_CT0_14** **Control Register 0 of GDMA Channel 14** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable

Bit(s)	Name	Description
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

100028EC **GDMA_CT1_14** **Control Register 1 of GDMA Channel 14** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RESERVED	CONT_MODE_EN	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_INT_EN	CH_UNMASK_FAIL_INT_EN	CH_MASK
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1

Bit(s)	Name	Description
		n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

100028F0 GDMA_SA_15 Source Address of GDMA Channel 15 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SOURCE_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SOURCE_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SOURCE_ADDR	**Souce address**

100028F4 GDMA_DA_15 Destination Address of GDMA Channel 15 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEST_ADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEST_ADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DEST_ADDR	**Destination address**

100028F8 GDMA_CT0_15 Control Register 0 of GDMA Channel 15 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TARGET_BYTE_CNT															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CURR_SEGMENT								SOURCE_ADDR_MODE	DEST_ADDR_MODE	BURST_SIZE			SEGMENT_DONE_INT_EN	CH_EN	SW_MODE_EN
Type	RO								RW	RW	RW			RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	TARGET_BYTE_CNT	**The number of bytes to be transferred**
15:8	CURR_SEGMENT	**Indicates the current segment (0 to 255)**
7	SOURCE_ADDR_MODE	**Sets the source address mode** 0: Incremental mode 1: Fix mode
6	DEST_ADDR_MODE	**Sets the destination address mode** 0: Incremental mode 1: Fix mode
5:3	BURST_SIZE	**Sets the number of double words in each burst transaction** 0: 1 DW 1: 2 DWs 2: 4 DWs 3: 8 DWs 4: 16 DWs 5: Undefined 6: Undefined 7: Undefined
2	SEGMENT_DONE_INT_EN	**Enable the segment done interrupt. This interrupt asserts after transfer of each segment is done.** 0: Disable 1: Enable
1	CH_EN	**If CONT_MODE_EN=0, this bit is de-asserted by hardware after the number of bytes transfferred reaches the TARGET_BYTE_CNT** 0: Disable 1: Enable
0	SW_MODE_EN	**Software mode enable. If software mode enable is set, the data transfer starts when the CH_EN bit is set. Otherwise, the data transfer starts when the DMA request is asserted.** 0: Hardware mode 1: Software mode

100028FC **GDMA_CT1_15** **Control Register 1 of GDMA Channel 15** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESERVED						NUM_SEGMENT				SOURCE_DMA_REQ					
Type	RO						RW				RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESERVED	CONT_MODE_	DEST_DMA_REQ						NEXT_CH2UNMASK					COHERENT_	CH_UNMAS	CH_MASK

		EN												INT_EN	K_FAIL_INT_EN	
Type	RO	RW	RW						RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
25:22	NUM_SEGMENT	**the number of segments=2N, where N is the value of this field. Valid values for this field are N=0 to 8. The segment size=(TARGET_BYTE_CNT/2N). It the TARGET_BYTE_CNT is not a multiple of 2N, the segment size = {(TARGET_BYTE_CNT/2N) + 1}.**
21:16	SOURCE_DMA_REQ	**Selects the source DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The source of the transfer is memory (always ready)
14	CONT_MODE_EN	**If CONT_MODE_EN=1, HW will NOT clear CH_EN after the number of bytes transferred reachs TARGET_BYTE_CNT. Otherwise, HW will clear CH_EN will clear the CH_EN.** 0: Continuous mode is disabled 1: Continuous mode is enabled
13:8	DEST_DMA_REQ	**Selects the destination DMA request** 0: DMA_REQ0 1: DMA_REQ1 2: DMA_REQ2 32: The destination of the transfer is memory (always ready)
7:3	NEXT_CH2UNMASK	**Selects the channel to clear the CH_MASK bit. When the number of bytes transferred reaches the TARGET_BYTE_CNT, the hardware will clear the CH_MASK field of the NEXT_CH2UNMASK channel. If the hardware does not need to clear CH_MASK field of any channel, this field should be set to the channel itself.** 0: Channel 0 1: Channel 1 n: Channel n
2	COHERENT_INT_EN	**If COHERENT_INT_EN is set, GDMA will issue a dummy read to destination after the last write to destination to avoid data coherent problem. Note: DO NOT set this field if the destination is not MEM. (may corrupt data, if destination is a FIFO)** 0: Disable 1: Enable
1	CH_UNMASK_FAIL_INT_EN	**If this field is set, an interrupt will be assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.** 0: Disable 1: Enable
0	CH_MASK	**When this field is set, the transfer of this channel is gated untill this field is clear by HW/SW.** 0: Channel is not masked 1: Channel is masked

10002A00	<u>GDMA_UNMASK_INTSTS</u>	Unmask Fail Interrupt Status	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	UNMASK_FAIL_INTSTS[31:16]															

Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**UNMASK_FAIL_INTSTS[15:0]**															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	UNMASK_FAIL_INTSTS	**This field is the bit-map of unmask fail interrupt status of each channel. The unmask fail interrupt will assert when HW detect the CH_MASK field of NEXT_CH2UNMASK channel is 1'b0 while trying to clear it.**

10002A04 **GDMA_DONE_INTSTS** **Segment Done Interrupt Status** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEGMENT_DONE_INTSTS[31:16]**															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEGMENT_DONE_INTSTS[15:0]**															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEGMENT_DONE_INTSTS	**This field is the bit-map of segment done interrupt status of each channel. The segment done interrupt will assert when each segment is transferred completely.**

10002A20 **GDMA_GCT** **Global Control** **0000000E**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESERVED[26:11]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESERVED[10:0]**											**TOTAL_CH_NUM**		**IP_VER**		**ARB_MODE**
Type	RO											RO		RO		RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit(s)	Name	Description
4:3	TOTAL_CH_NUM	**Total channel number supported** 0: 8 channels 1: 16 channels 2: 32 channels 3: Undefined
2:1	IP_VER	**GDMA core version**
0	ARB_MODE	**Arbitration mode selection** 0: channel 0 has highest priority and others are round-robin 1: All channel are round-robin

10002A30	GDMA_PERI_ADDR_START_0	Peripheral Region 0 Starting Address	10000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERI_ADDR_START_0[31:16]															
Type	RW															
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PERI_ADDR_START_0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_START_0	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A34	GDMA_PERI_ADDR_END_0	Peripheral Region 0 End Address	20000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERI_ADDR_END_0[31:16]															
Type	RW															
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PERI_ADDR_END_0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_END_0	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A38	GDMA_PERI_ADDR_START_1	Peripheral Region 1 Starting Address	20000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERI_ADDR_START_1[31:16]															
Type	RW															
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PERI_ADDR_START_1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_START_1	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A3C	GDMA_PERI_ADDR_END_1	Peripheral Region 1 End Address	30000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERI_ADDR_END_1[31:16]															

Type	RW															
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PERI_ADDR_END_1[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_END_1	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A40	**GDMA_PERI_ADDR_START_2**	**Peripheral Region 2 Starting Address**	**10000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PERI_ADDR_START_2[31:16]**															
Type	RW															
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PERI_ADDR_START_2[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_START_2	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A44	**GDMA_PERI_ADDR_END_2**	**Peripheral Region 2 End Address**	**20000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PERI_ADDR_END_2[31:16]**															
Type	RW															
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PERI_ADDR_END_2[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_END_2	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A48	**GDMA_PERI_ADDR_START_3**	**Peripheral Region 3 Starting Address**	**60000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PERI_ADDR_START_3[31:16]**															
Type	RW															
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PERI_ADDR_START_3[15:0]**															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_START_3	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

10002A4C **GDMA_PERI_ADDR_END_3** **Peripheral Region 3 End Address** **70000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERI_ADDR_END_3[31:16]															
Type	RW															
Reset	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PERI_ADDR_END_3[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERI_ADDR_END_3	**GDMA request will direct to peripheral bus if the request address >= PERI_ADDR_START_x & < PERI_ADDR_END_x**

2.16 AES Controller

2.16.1 Registers

AES Changes LOG

Revision	Date	Author	Change Log
0.1	2013/4/30	Morrie Lin	Initialization
0.2	2013/6/5	Morrie Lin	Add desc_5dw_info_en register
0.3	2013/6/7	Morrie Lin	Update AES base address

Module name: AES Base address: (+10004000h)

Address	Name	Width	Register Function
10004000	TX_BASE_PTR0	32	**TX_BASE_PTR0** Used for DMA base address of TX ring0
10004004	TX_MAX_CNT0	32	**TX_MAX_CNT0** Used for DMA max number of TX ring0
10004008	TX_CTX_IDX0	32	**TX_CTX_IDX0** Used for CPU pointer of TX ring0
1000400C	TX_DTX_IDX0	32	**TX_DTX_IDX0** Used for DMA pointer of TX ring0
10004100	RX_BASE_PTR0	32	**RX_BASE_PTR0** Used for DMA base address of RX ring0
10004104	RX_MAX_CNT0	32	**RX_MAX_CNT0** Used for DMA max number of RX ring0
10004108	RX_CALC_IDX0	32	**RX_CALC_IDX0** Used for CPU pointer of RX ring0
1000410C	FS_DRX_IDX0	32	**FS_DRX_IDX0** Used for DMA pointer of RX ring0
10004200	PDMA_INFO	32	**PDMA_INFO** used for PDMA information
10004204	PDMA_GLO_CFG	32	**PDMA_GLO_CFG** used for PDMA setting
10004208	PDMA_RST_IDX	32	**PDMA_RST_IDX** used for PDMA setting
1000420C	DELAY_INT_CFG	32	**DELAY_INT_CFG** used for PDMA setting
10004210	PDMA_Q_CFG	32	**PDMA_Q_CFG** used for PDMA setting
10004220	PDMA_INT_STA	32	**PDMA_INT_STA** used for PDMA setting
10004228	PDMA_INT_MSK	32	**PDMA_INT_MSK** used for PDMA setting

10004000 **TX_BASE_PTR0** **TX_BASE_PTR0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TX_BASE_PTR0**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**Reserved**
15:0	TX_BASE_PTR0	**Tx Base Pointer 0** Points to the base address of TX_Ring 0 (If enable desc_5dw_info_en 8-DWORD aligned address, else 4-DWORD aligned address).

10004004 **TX_MAX_CNT0** **TX_MAX_CNT0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TX_MAX_CNT0**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**Reserved**
15:0	TX_MAX_CNT0	**Tx Maximum TXD Count 0** The maximum TXD count in TXD_Ring 0.

10004008 **TX_CTX_IDX0** **TX_CTX_IDX0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TX_MAX_CNT0**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**Reserved**
15:0	TX_MAX_CNT0	**Tx CPU TXD Index n** Points to the next TXD to be used by the CPU. (If enable desc_5dw_info_en, 8-DWORD aligned address, else 4-DWORD aligned address).

1000400C **TX_DTX_IDX0** **TX_DTX_IDX0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV[23:8]**															

Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[7:0]								TX_DTX_IDX0							
Type	RO								RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:8	RESV	**Reserved**
7:0	TX_DTX_IDX0	**Tx DMA TXD Index n** Points to the next TXD to be used by the DMA. (If enable desc_5dw_info_en, 8-DWORD aligned address, else 4-DWORD aligned address).

10004100 **RX_BASE_PTR0** **RX_BASE_PTR0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_BASE_PTR0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**Reserved**
15:0	RX_BASE_PTR0	**Rx Base Pointer 0** Points to the base address of RXD Ring 0 (If enable desc_5dw_info_en, 8-DWORD aligned address, else 4-DWORD aligned address).

10004104 **RX_MAX_CNT0** **RX_MAX_CNT0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_MAX_CNT0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**Reserved**
15:0	RX_MAX_CNT0	**Rx Maximum Count 0** The maximum RXD count in RXD Ring 0.

10004108 **RX_CALC_IDX0** **RX_CALC_IDX0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_CALI_IDX0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**Reserved**
15:0	RX_CALI_IDX0	**Rx CPU RXD Index 0** Points to the next RXD the CPU will allocate to RXD Ring 0. (If enable desc_5dw_info_en, 8-DWORD aligned address, else 4-DWORD aligned address).

1000410C **FS_DRX_IDX0** **FS_DRX_IDX0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[23:8]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[7:0]								RX_DRX_IDX0							
Type	RO								RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:8	RESV	**Reserved**
7:0	RX_DRX_IDX0	**Rx DMA RXD Index n** Points to the next RXD that the DMA will use in FDS Ring 0. (If enable desc_5dw_info_en, 8-DWORD aligned address, else 4-DWORD aligned address).

10004200 **PDMA_INFO** **PDMA_INFO** **4C000101**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VERSION				INDEX_WIDTH				BASE_PTR_WIDTH							
Type	RO				RO				RO							
Reset	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_RING_NUM								TX_RING_NUM							
Type	RO								RO							
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:28	VERSION	**PDMA controller version.**
27:24	INDEX_WIDTH	**RX Ring index width**
23:16	BASE_PTR_WIDTH	**Base Pointer Width**
15:8	RX_RING_NUM	**Rx ring number**
7:0	TX_RING_NUM	**Tx ring number**

10004204	PDMA_GLO_CFG	PDMA_GLO_CFG	00000450

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_2B_OFFSET	CLKGATE_BYP	BYTE_SWAP	RESV[16:4]												
Type	RW	RO	RO	RO												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[3:0]				desc_5dw_info_en	multi_dma_en	share_fifo_en	desc_32b_en	BIG_ENDIAN	TX_WB_DDONE	WPDMA_BT_SIZE		RX_DMA_BUSY	RX_DMA_EN	TX_DMA_BUSY	TX_DMA_EN
Type	RO				RW	RW	RW	RW	RW	RW	RW		RO	RW	RO	RW
Reset	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0

Bit(s)	Name	Description
31	RX_2B_OFFSET	**Rx 2 Byte Offset** Sets the byte size of the Rx buffer offset. 0: 4 bytes 1: 2 bytes. 0
30	CLKGATE_BYP	**Clock Gating Control Status Register** Controls gating of the PDMA clock. 0: PDMA clock operates in freerun mode. 1: PDMA clock is gated when idle.
29	BYTE_SWAP	**Byte Swap** The DMA applies the endian rule to convert the descriptor. 0: Byte swap not applied. 1: Apply byte swap.
28:12	RESV	**Reserved**
11	desc_5dw_info_en	**Support extension tx_info/rx_info to to 20 byte and the total length of descriptor is 32 byte.** 0: Disable 1: Enable
10	multi_dma_en	
9	share_fifo_en	
8	desc_32b_en	**Support 32 Byte alignment descriptor** Enables support for 32 Byte alignment PDMA descriptors. 0: Disable 1: Enable
7	BIG_ENDIAN	**Selects the Endian mode for the SoC platform section.** DMA applies the endian rule to convert payload and Tx/Rx information. DMA does not apply the endian rule to registers or descriptors. 0: Little endian 1: Big endian
6	TX_WB_DDONE	**Tx Write Back DDONE** Enables TX_DMA writing back DDONE into TXD. 0: Disable 1: Enable
5:4	WPDMA_BT_SIZE	**PDMA Burst Size** Defines the burst size of PDMA. 0 : 4 DWORD (16bytes). 1 : 8 DWORD (32 bytes). 2 : 16 DWORD (64 bytes).

Bit(s)	Name	Description
		3 : 32 DWORD (128 bytes)
3	RX_DMA_BUSY	**1 : RX_DMA is busy. 0 : RX_DMA is not busy**
2	RX_DMA_EN	**Rx DMA Enable** Enables Rx DMA. When disabled, Rx DMA finishes the current receiving packet, and then stops. 0: Disable 1: Enable
1	TX_DMA_BUSY	**Indicates whether Tx DMA is busy.** 0: Not busy 1: Busy
0	TX_DMA_EN	**Tx DMA Enable** Enables Tx DMA. When disabled, Tx DMA finishes the current sending packet, and then stops. 0: Disable 1: Enable

10004208 **PDMA_RST_IDX** **PDMA_RST_IDX** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RESV	**Reserved**

1000420C **DELAY_INT_CFG** **DELAY_INT_CFG** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TXDLY_INT_EN	TXMAX_PINT							TXMAX_PTIME							
Type	RW	RW							RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RXDLY_INT_EN	RXMAX_PINT							RXMAX_PTIME							
Type	RW	RW							RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	TXDLY_INT_EN	**Tx Delay Interrupt Enable** Enables the Tx delayed interrupt mechanism. 0: Disable

Bit(s)	Name	Description
		1: Enable
30:24	TXMAX_PINT	**Tx Maximum Pending Interrupts** Specifies the maximum number of pending interrupts. When the number of pending interrupts is equal to or greater than the value specified here or the interrupt pending time has reached the limit (see below), a final TX_DLY_INT is generated. 0: Disable this feature.
23:16	TXMAX_PTIME	**Tx Maximum Pending Time** Specifies the maximum pending time for the internal TX_DONE_INT0 and TX_DONE_INT1. When the pending time is equal to or greater than TXMAX_PTIME x 20us or the number of pended TX_DONE_INT0 and TX_DONE_INT1 is equal to or greater than TXMAX_PINT (see above), a final TX_DLY_INT is generated 0: Disable this feature.
15	RXDLY_INT_EN	**Rx Delay Interrupt Enable** Enables the Rx delayed interrupt mechanism. 0: Disable 1: Enable
14:8	RXMAX_PINT	**Rx Maximum Pending Interrupts** Specifies the maximum number of pending interrupts. When the number of pended interrupts is equal to or greater than the value specified here or the interrupt pending time has reached the limit (see below), a final RX_DLY_INT is generated. 0: Disable this feature.
7:0	RXMAX_PTIME	**Rx Maximum Pending Time** Specifies the maximum pending time for the internal RX_DONE_INT. When the pending time is equal to or greater than RXMAX_PTIME x 20 us, or the number of pended RX_DONE_INT is equal to or greater than RXMAX_PCNT (see above), a final RX_DLY_INT is generated. 0: Disable this feature.

10004210 **PDMA_Q_CFG** **PDMA_Q_CFG** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[11:0]												RST_DRX_IDX1			
Type	RO												RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RESV	
3:0	RST_DRX_IDX1	**Will stop to block interface as RX-descriptors reach this threshold**

10004220 **PDMA_INT_STA** **PDMA_INT_STA** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_COHERE	RX_DLY_INT	TX_COHERE	TX_DLY_INT	RESV1											RX_DONE_INT

	NT		NT													
Type	RW	RW	RW	RW	RO											RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV															TX_DONE_INT
Type	RO															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	RX_COHERENT	**Rx Coherent Interrupt** Asserts when the Rx DMA is ready to handle a queue, but cannot access the queue because the driver is not ready.
30	RX_DLY_INT	**Rx Delay Interrupt** Asserts when the number of pended Rx interrupts has reached a specified level, or when the pending time is reached. Configure this interrupt using the DELAY_INT_CFG register.
29	TX_COHERENT	**Tx Coherent Interrupt** Asserts when the Tx DMA is ready to handle a queue, but cannot access the queue because the driver is not ready.
28	TX_DLY_INT	**Tx Delay Interrupt** Asserts when the number of pended Tx interrupts has reached a specified level, or when the pending time is reached. Configure this interrupt using the DELAY_INT_CFG register.
27:17	RESV1	
16	RX_DONE_INT	**Rx Queue 0 Done Interrupt** Asserts when an Rx packet is received on Queue 0.
15:1	RESV	
0	TX_DONE_INT	**Tx Queue 0 Done Interrupt** Asserts when a Tx Queue 0 packet is transmitted.

10004228 **PDMA_INT_MSK** **PDMA_INT_MSK** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_COHERENT_EN	RX_DLY_INT_EN	TX_COHERENT_INT_EN	TX_DLY_INT_EN	RESV1											RX_DONE_INT_EN
Type	RW	RW	RW	RW	RO											RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV															TX_DONE_INT_EN
Type	RO															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	RX_COHERENT_EN	**Masks the Rx Coherent interrupt. This interrupt asserts when the Rx DMA is ready to handle a queue, but cannot access the queue because**

Bit(s)	Name	Description
		the driver is not ready.
30	RX_DLY_INT_EN	**Masks the Rx Delay interrupt. This interrupt asserts when the number of pending Rx interrupts has reached a specified level, or when the pending time is reached.**
29	TX_COHERENT_INT_EN	**Masks the Tx Coherent interrupt. This interrupt asserts when the Tx DMA is ready to handle a queue, but cannot access the queue because the driver is not ready.**
28	TX_DLY_INT_EN	**Masks the Tx Delay interrupt. This interrupt asserts when the number of pending Tx interrupts has reached a specified level, or when the pending time is reached.**
27:17	RESV1	
16	RX_DONE_INT_EN	**Masks the Rx Queue 0 Done interrupt. This interrupt asserts when an Rx packet is received on Queue 0.**
15:1	RESV	
0	TX_DONE_INT_EN	**Masks the Tx Queue 0 Done interrupt. This interrupt asserts when a Tx packet is transmitted on Queue 0.**

2.17 PWM (Pulse Width Modulation)

2.17.1 Registers

PWM Changes LOG

Revision	Date	Author	Change Log
1	2013/11/26	Rick Ho	Initial Version

Module name: PWM Base address: (+10005000h)

Address	Name	Width	Register Function
10005000	PWM_ENABLE	32	PWM Enable register
10005010	PWM0_CON	32	PWM0 Control register
10005014	PWM0_HDURATION	32	PWM0 High Duration register
10005018	PWM0_LDURATION	32	PWM0 Low Duration register
1000501C	PWM0_GDURATION	32	PWM0 Guard Duration register
10005030	PWM0_SEND_DATA0	32	PWM0 Send Data0 register
10005034	PWM0_SEND_DATA1	32	PWM0 Send Data1 register
10005038	PWM0_WAVE_NUM	32	PWM0 Wave Number register
1000503C	PWM0_DATA_WIDTH	32	PWM0 Data Width register
10005040	PWM0_THRESH	32	PWM0 Thresh register
10005044	PWM0_SEND_WAVENUM	32	PWM0 Send Wave Number register
10005050	PWM1_CON	32	PWM1 Control register
10005054	PWM1_HDURATION	32	PWM1 High Duration register
10005058	PWM1_LDURATION	32	PWM1 Low Duration register
1000505C	PWM1_GDURATION	32	PWM1 Guard Duration register
10005070	PWM1_SEND_DATA0	32	PWM1 Send Data0 register
10005074	PWM1_SEND_DATA1	32	PWM1 Send Data1 register
10005078	PWM1_WAVE_NUM	32	PWM1 Wave Number register
1000507C	PWM1_DATA_WIDTH	32	PWM1 Data Width register
10005080	PWM1_THRESH	32	PWM1 Thresh register
10005084	PWM1_SEND_WAVENUM	32	PWM1 Send Wave Number register
10005090	PWM2_CON	32	PWM2 Control register
10005094	PWM2_HDURATION	32	PWM2 High Duration register

10005098	PWM2_LDURATION	32	**PWM2 Low Duration register**
1000509C	PWM2_GDURATION	32	**PWM2 Guard Duration register**
100050B0	PWM2_SEND_DATA0	32	**PWM2 Send Data0 register**
100050B4	PWM2_SEND_DATA1	32	**PWM2 Send Data1 register**
100050B8	PWM2_WAVE_NUM	32	**PWM2 Wave Number register**
100050BC	PWM2_DATA_WIDTH	32	**PWM2 Data Width register**
100050C0	PWM2_THRESH	32	**PWM2 Thresh register**
100050C4	PWM2_SEND_WAVENUM	32	**PWM2 Send Wave Number register**
100050D0	PWM3_CON	32	**PWM3 Control register**
100050D4	PWM3_HDURATION	32	**PWM3 High Duration register**
100050D8	PWM3_LDURATION	32	**PWM3 Low Duration register**
100050DC	PWM3_GDURATION	32	**PWM3 Guard Duration register**
100050F0	PWM3_SEND_DATA0	32	**PWM3 Send Data0 register**
100050F4	PWM3_SEND_DATA1	32	**PWM3 Send Data1 register**
100050F8	PWM3_WAVE_NUM	32	**PWM3 Wave Number register**
100050FC	PWM3_DATA_WIDTH	32	**PWM3 Data Width register**
10005100	PWM3_THRESH	32	**PWM3 Thresh register**
10005104	PWM3_SEND_WAVENUM	32	**PWM3 Send Wave Number register**
1000520C	PWM_EN_STATUS	32	**PWM Enable Status register**

10005000 **PWM_ENABLE** **PWM Enable register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[11:0]												PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN
Type	RO												RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RESV	**RESERVED**
3	PWM3_EN	0: disabe PWM3 1: enable PWM3

Bit(s)	Name	Description
2	PWM2_EN	0: disable PWM2 1: enable PWM2
1	PWM1_EN	0: disable PWM1 1: enable PWM1
0	PWM0_EN	0: disable PWM0 1: enable PWM0

10005010 **PWM0_CON** **PWM0 Control register** **00007E00**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS						GUARD_VALUE	IDLE_VALUE	RESV1			CLKSEL	CLKDIV		
Type	RW	RW						RW	RW	RO			RW	RW		
Reset	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV0	**RESERVED**
15	OLD_PWM_MODE	0: New PWM mode 1: Old PWM mode
14:9	STOP_BITPOS	**The stop bit position for source data in periodical mode. In FIFO mode, it's used to indicate the stop bit position in total 64 bits. In Memory mode, it's for the stop bit position in the last 32 bits.**
8	GUARD_VALUE	**PWM0 output value when guard time.**
7	IDLE_VALUE	**PWM0 output value when idle state.**
6:4	RESV1	**RESERVED**
3	CLKSEL	**Select PWM0 clock** 0: CLK= 100KHz 1: CLK= 40MHz
2:0	CLKDIV	**Select PWM0 clock scale.** 000: CLK Hz 001: CLK/2 Hz 010: CLK/4 Hz 011: CLK/8 Hz 100: CLK/16 Hz 101: CLK/32 Hz 110: CLK/64 Hz 111: CLK/128 Hz

10005014 **PWM0_HDURATION** **PWM0 High Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	HDURATION	**PWM0 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

10005018 **PWM0_LDURATION** **PWM0 Low Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	LDURATION	**PWM0 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

1000501C **PWM0_GDURATION** **PWM0 Guard Duration register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	GUARD_DURATION	

10005030 **PWM0_SEND_DATA0** **PWM0 Send Data0 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	SEND_DATA0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA0	**PWM0 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

10005034 **PWM0_SEND_DATA1** **PWM0 Send Data1 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA1	**PWM0 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

10005038 **PWM0_WAVE_NUM** **PWM0 Wave Number register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_NUM															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	WAVE_NUM	**The number by which PWM0 will generate from the pulse data repeatedly.** Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

1000503C **PWM0_DATA_WIDTH** **PWM0 Data Width register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[18:3]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[2:0]			DATA_WIDTH												

Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	DATA_WIDTH	**The PWM0 pulse data width in the old PWM mode.**

10005040 **PWM0_THRESH** **PWM0 Thresh register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV[18:3]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESV[2:0]**			**THRESH**												
Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	THRESH	**The PWM0 pulse data high/low switching threshold in the old PWM mode.**

10005044 **PWM0_SEND_WAVENUM** **PWM0 Send Wave Number register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_WAVENUM**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	SEND_WAVENUM	**The number by which PWM0 has already generated from the specified data source in the periodical mode.**

10005050 **PWM1_CON** **PWM1 Control register** **00007E00**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV0**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**OLD_PWM_MODE**	**STOP_BITPOS**						**GUARD_VALUE**	**IDLE_VALUE**	**RESV1**			**CLKSEL**	**CLKDIV**		

Type	RW	RW						RW	RW	RO			RW	RW		
Reset	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV0	**RESERVED**
15	OLD_PWM_MODE	**Use old PWM mode** Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source (however could not work in the system sleep-mode). 0: New PWM mode 1: Old PWM mode
14:9	STOP_BITPOS	**Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source (however could not work in the system sleep-mode).**
8	GUARD_VALUE	**PWM1 output value when guard time.**
7	IDLE_VALUE	**PWM1 output value when idle state.**
6:4	RESV1	**Select Random Generator mode**
3	CLKSEL	**Select PWM1 clock** 0: CLK= 100KHz 1: CLK= 40MHz
2:0	CLKDIV	**Select PWM1 clock scale.** 000: CLK Hz 001: CLK/2 Hz 010: CLK/4 Hz 011: CLK/8 Hz 100: CLK/16 Hz 101: CLK/32 Hz 110: CLK/64 Hz 111: CLK/128 Hz

10005054 **PWM1_HDURATION** **PWM1 High Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	HDURATION	**PWM1 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

10005058 **PWM1_LDURATION** **PWM1 Low Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**LDURATION**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	LDURATION	**PWM1 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

10005 05C **PWM1_GDURATION** **PWM1 Guard Duration register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GUARD_DURATION**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	GUARD_DURATION	

10005070 **PWM1_SEND_DATA0** **PWM1 Send Data0 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA0[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_DATA0[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA0	**PWM1 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

10005074 **PWM1_SEND_DATA1** **PWM1 Send Data1 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SEND_DATA1[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	**SEND_DATA1[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA1	**PWM1 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

10005078 **PWM1_WAVE_NUM** **PWM1 Wave Number register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WAVE_NUM**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	WAVE_NUM	**The number by which PWM1 will generate from the pulse data repeatedly.** Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

1000507C **PWM1_DATA_WIDTH** **PWM1 Data Width register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV[18:3]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESV[2:0]**			**DATA_WIDTH**												
Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	DATA_WIDTH	**The PWM1 pulse data width in the old PWM mode.**

10005080 **PWM1_THRESH** **PWM1 Thresh register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV[18:3]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESV[2:0]**			**THRESH**												
Type	RO			RW												

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	THRESH	**The PWM1 pulse data high/low switching threshold in the old PWM mode.**

10005084 **PWM1_SEND_WAVENUM** **PWM1 Send Wave Number register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SEND_WAVENUM**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	SEND_WAVENUM	**The number by which PWM1 has already generated from the specified data source in the periodical mode.**

10005090 **PWM2_CON** **PWM2 Control register** **00007E00**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESV0**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**OLD_PWM_MODE**	**STOP_BITPOS**						**GUARD_VALUE**	**IDLE_VALUE**	**RESV1**			**CLKSEL**	**CLKDIV**		
Type	RW	RW						RW	RW	RO			RW	RW		
Reset	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV0	**RESERVED**
15	OLD_PWM_MODE	**Use old PWM mode** Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source (however could not work in the system sleep-mode). 0: New PWM mode 1: Old PWM mode
14:9	STOP_BITPOS	**Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source (however could not work in the system sleep-mode).**
8	GUARD_VALUE	**PWM2 output value when guard time.**
7	IDLE_VALUE	**PWM2 output value when idle state.**
6:4	RESV1	**Select Random Generator mode**

Bit(s)	Name	Description
3	CLKSEL	**Select PWM2 clock** 0: CLK= 100KHz 1: CLK= 40MHz
2:0	CLKDIV	**Select PWM2 clock scale.** 000: CLK Hz 001: CLK/2 Hz 010: CLK/4 Hz 011: CLK/8 Hz 100: CLK/16 Hz 101: CLK/32 Hz 110: CLK/64 Hz 111: CLK/128 Hz

10005094 **PWM2_HDURATION** **PWM2 High Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	HDURATION	**PWM2 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

10005098 **PWM2_LDURATION** **PWM2 Low Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	LDURATION	**PWM2 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

1000509C **PWM2_GDURATION** **PWM2 Guard Duration register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	GUARD_DURATION	

100050B0 PWM2_SEND_DATA0 PWM2 Send Data0 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA0	**PWM2 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

100050B4 PWM2_SEND_DATA1 PWM2 Send Data1 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA1	**PWM2 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

100050B8 PWM2_WAVE_NUM PWM2 Wave Number register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	WAVE_NUM															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	WAVE_NUM	**The number by which PWM2 will generate from the pulse data repeatedly.** Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

100050BC **PWM2_DATA_WIDTH** **PWM2 Data Width register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[18:3]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[2:0]			DATA_WIDTH												
Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	DATA_WIDTH	**The PWM2 pulse data width in the old PWM mode.**

100050C0 **PWM2_THRESH** **PWM2 Thresh register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[18:3]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[2:0]			THRESH												
Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	THRESH	**The PWM2 pulse data high/low switching threshold in the old PWM mode.**

100050C4 **PWM2_SEND_WAVENUM** **PWM2 Send Wave Number register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	SEND_WAVENUM	**The number by which PWM2 has already generated from the specified data source in the periodical mode.**

100050D0 **PWM3_CON** **PWM3 Control register** **00007E00**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OLD_PWM_MODE	STOP_BITPOS						GUARD_VALUE	IDLE_VALUE	RESV1			CLKSEL	CLKDIV		
Type	RW	RW						RW	RW	RO			RW	RW		
Reset	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV0	**RESERVED**
15	OLD_PWM_MODE	**Use old PWM mode** Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source (however could not work in the system sleep-mode). 0: New PWM mode 1: Old PWM mode
14:9	STOP_BITPOS	**Note: Using old PWM mode also means periodical mode. So SRCSEL and MODE is ignored in this situation. Only old PWM mode with 32 KHz clock source (however could not work in the system sleep-mode).**
8	GUARD_VALUE	**PWM3 output value when guard time.**
7	IDLE_VALUE	**PWM3 output value when idle state.**
6:4	RESV1	**Select Random Generator mode**
3	CLKSEL	**Select PWM3 clock** 0: CLK= 100KHz 1: CLK= 40MHz
2:0	CLKDIV	**Select PWM3 clock scale.** 000: CLK Hz 001: CLK/2 Hz 010: CLK/4 Hz 011: CLK/8 Hz 100: CLK/16 Hz 101: CLK/32 Hz 110: CLK/64 Hz 111: CLK/128 Hz

100050D4 **PWM3_HDURATION** **PWM3 High Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	HDURATION	**PWM3 pulse duration based on the current clock when PWM output is high. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

100050D8 **PWM3_LDURATION** **PWM3 Low Duration register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LDURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	LDURATION	**PWM3 pulse duration based on the current clock when PWM output is low. If duration =N, need to program N-1 in this register.** Note: The duration of PWM must not be 0.

100050DC **PWM3_GDURATION** **PWM3 Guard Duration register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GUARD_DURATION															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	GUARD_DURATION	

100050F0 **PWM3_SEND_DATA0** **PWM3 Send Data0 register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA0[31:16]															

Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA0	**PWM3 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

100050F4 PWM3_SEND_DATA1 PWM3 Send Data1 register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEND_DATA1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_DATA1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	SEND_DATA1	**PWM3 local buffer0 of pulse sequence data to be generated.** Note: This value should be written only in periodically FIFO mode. In other mode, this buffer is for internal memory access.

100050F8 PWM3_WAVE_NUM PWM3 Wave Number register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WAVE_NUM															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	WAVE_NUM	**The number by which PWM3 will generate from the pulse data repeatedly.** Note: If WAVE_NUM=0, the waveform generation will not stop until it is disabled.

100050FC PWM3_DATA_WIDTH PWM3 Data Width register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[18:3]															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[2:0]			DATA_WIDTH												
Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	DATA_WIDTH	**The PWM3 pulse data width in the old PWM mode.**

10005100 **PWM3_THRESH** **PWM3 Thresh register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[18:3]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[2:0]			THRESH												
Type	RO			RW												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:13	RESV	**RESERVED**
12:0	THRESH	**The PWM3 pulse data high/low switching threshold in the old PWM mode.**

10005104 **PWM3_SEND_WAVENUM** **PWM3 Send Wave Number register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEND_WAVENUM															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	RESV	**RESERVED**
15:0	SEND_WAVENUM	**The number by which PWM3 has already generated from the specified data source in the periodical mode.**

1000520C **PWM_EN_STATUS** **PWM Enable Status register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESV[27:12]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESV[11:0]												PWM3_	PWM2_	PWM1_	PWM0_

													EN_ ST	**EN_ ST**	**EN_ ST**	**EN_ ST**
Type	RO												RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:4	RESV	**RESERVED**
3	PWM3_EN_ST	**PWM3 enable status**
2	PWM2_EN_ST	**PWM2 enable status**
1	PWM1_EN_ST	**PWM1 enable status**
0	PWM0_EN_ST	**PWM0 enable status**

2.18 Frame Engine

2.18.1 Registers

SDM Changes LOG

Revision	Date	Author	Change Log
0.1	2013/5/27	PeterCT WU	Initialization

Module name: SDM Base address: (+10100000h)

Address	Name	Width	Register Function
10100800	TX_BASE_PTR_0	32	TX Ring #0 Base Pointer
10100804	TX_MAX_CNT_0	32	TX Ring #0 Maximum Count
10100808	TX_CTX_IDX_0	32	TX Ring #0 CPU pointer
1010080C	TX_DTX_IDX_0	32	TX Ring #0 DMA poitner
10100810	TX_BASE_PTR_1	32	TX Ring #1 Base Pointer
10100814	TX_MAX_CNT_1	32	TX Ring #1 Maximum Count
10100818	TX_CTX_IDX_1	32	TX Ring #1 CPU pointer
1010081C	TX_DTX_IDX_1	32	TX Ring #1 DMA poitner
10100820	TX_BASE_PTR_2	32	TX Ring #2 Base Pointer
10100824	TX_MAX_CNT_2	32	TX Ring #2 Maximum Count
10100828	TX_CTX_IDX_2	32	TX Ring #2 CPU pointer
1010082C	TX_DTX_IDX_2	32	TX Ring #2 DMA poitner
10100830	TX_BASE_PTR_3	32	TX Ring #3 Base Pointer
10100834	TX_MAX_CNT_3	32	TX Ring #3 Maximum Count
10100838	TX_CTX_IDX_3	32	TX Ring #3 CPU pointer
1010083C	TX_DTX_IDX_3	32	TX Ring #3 DMA poitner
10100900	RX_BASE_PTR_0	32	RX Ring #0 Base Pointer
10100904	RX_MAX_CNT_0	32	RX Ring #0 Maximum Count
10100908	RX_CRX_IDX_0	32	RX Ring #0 CPU pointer
1010090C	RX_DRX_IDX_0	32	RX Ring #0 DMA poitner
10100910	RX_BASE_PTR_1	32	RX Ring #1 Base Pointer
10100914	RX_MAX_CNT_1	32	RX Ring #1 Maximum Count
10100918	RX_CRX_IDX_1	32	RX Ring #1 CPU pointer
1010091C	RX_DRX_IDX_1	32	RX Ring #1 DMA poitner
10100A00	PDMA_INFO	32	PDMA Information
10100A04	PDMA_GLO_CFG	32	PDMA Global Configuration
10100A0C	DELAY_INT_CF	32	Delay Interrupt Configuration

	G		
10100A10	FREEQ_THRES	32	Free Queue Threshold
10100A20	INT_STATUS	32	Interrupt Status
10100A28	INT_MASK	32	Interrupt Mask
10100A80	PDMA_SCH	32	Scheduler Configuration for Q0&Q1
10100A84	PDMA_WRR	32	Scheduler Configuration for Q2&Q3
10100C00	SDM_CON	32	Switch DMA Control
10100C04	SDM_RING	32	Switch DMA Rx Ring
10100C08	SDM_TRING	32	Switch DMA TX Ring
10100C0C	SDM_MAC_ADRL	32	Switch MAC Address LSB
10100C10	SDM_MAC_ADRH	32	Switch MAC Address MSB
10100D00	SDM_TPCNT	32	Switch DMA Tx Packet Count
10100D04	SDM_TBCNT	32	Switch DMA TX Byte Count
10100D08	SDM_RPCNT	32	Switch DMA RX Packet Count
10100D0C	SDM_RBCNT	32	Switch DMA RX Byte Count
10100D10	SDM_CS_ERR	32	Switch DMA RX Checksum Error

10100800 **TX_BASE_PTR_0** **TX Ring #0 Base Pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TX_BASE_PTR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TX_BASE_PTR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_BASE_PTR	Point to the base address of TX Ring #0 (4-DW aligned address)

10100804 **TX_MAX_CNT_0** **TX Ring #0 Maximum Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_MAX_CNT											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_MAX_CNT	The maximum number of TXD count in TX Ring #0

10100808 **TX_CTX_IDX_** **TX Ring #0 CPU pointer** **0000000**

	0		**0**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_CTX_IDX											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_CTX_IDX	**Point to the next TXD CPU wants to use**

1010080C	TX_DTX_IDX_0	TX Ring #0 DMA poitner	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_DTX_IDX											
Type					RO											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_DTX_IDX	**Point to the next TXD DMA wants to use**

10100810	TX_BASE_PTR_1	TX Ring #1 Base Pointer	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TX_BASE_PTR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TX_BASE_PTR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_BASE_PTR	**Point to the base address of TX Ring #0 (4-DW aligned address)**

10100814	TX_MAX_CNT_1	TX Ring #1 Maximum Count	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_MAX_CNT											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_MAX_CNT	**The maximum number of TXD count in TX Ring #0**

10100818 **TX_CTX_IDX_1** **TX Ring #1 CPU pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_CTX_IDX											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_CTX_IDX	**Point to the next TXD CPU wants to use**

1010081C **TX_DTX_IDX_1** **TX Ring #1 DMA poitner** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_DTX_IDX											
Type					RO											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_DTX_IDX	**Point to the next TXD DMA wants to use**

10100820 **TX_BASE_PTR_2** **TX Ring #2 Base Pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TX_BASE_PTR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TX_BASE_PTR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_BASE_PTR	**Point to the base address of TX Ring #0 (4-DW aligned address)**

10100824 **TX_MAX_CNT_2** **TX Ring #2 Maximum Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**TX_MAX_CNT**											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_MAX_CNT	**The maximum number of TXD count in TX Ring #0**

10100828 **TX_CTX_IDX_2** **TX Ring #2 CPU pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**TX_CTX_IDX**											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_CTX_IDX	**Point to the next TXD CPU wants to use**

1010082C **TX_DTX_IDX_2** **TX Ring #2 DMA poitner** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					**TX_DTX_IDX**											
Type					RO											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_DTX_IDX	**Point to the next TXD DMA wants to use**

10100830 **TX_BASE_PTR_3** **TX Ring #3 Base Pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**TX_BASE_PTR[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TX_BASE_PTR[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_BASE_PTR	**Point to the base address of TX Ring #0 (4-DW aligned address)**

10100834 **TX_MAX_CNT_3** **TX Ring #3 Maximum Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_MAX_CNT											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_MAX_CNT	**The maximum number of TXD count in TX Ring #0**

10100838 **TX_CTX_IDX_3** **TX Ring #3 CPU pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_CTX_IDX											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_CTX_IDX	**Point to the next TXD CPU wants to use**

1010083C **TX_DTX_IDX_3** **TX Ring #3 DMA poitner** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_DTX_IDX											
Type					RO											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_DTX_IDX	**Point to the next TXD DMA wants to use**

10100900 **RX_BASE_PTR_0** **RX Ring #0 Base Pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_BASE_PTR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	RX_BASE_PTR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RX_BASE_PTR	**Point to the base address of RX Ring #0 (4-DW aligned address)**

10100904 **RX_MAX_CNT_0** **RX Ring #0 Maximum Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_MAX_CNT											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_MAX_CNT	**The maximum number of RXD count in RX Ring #0**

10100908 **RX_CRX_IDX_0** **RX Ring #0 CPU pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_CTX_IDX											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_CTX_IDX	**Point to the next RXD CPU wants to use**

1010090C **RX_DRX_IDX_0** **RX Ring #0 DMA poitner** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					RX_DRX_IDX											
Type					RO											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	RX_DRX_IDX	**Point to the next RXD DMA wants to use**

10100910 **RX_BASE_PTR_1** **RX Ring #1 Base Pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_BASE_PTR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_BASE_PTR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RX_BASE_PTR	**Point to the base address of RX Ring #0 (4-DW aligned address)**

10100914 **RX_MAX_CNT_1** **RX Ring #1 Maximum Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_MAX_CNT											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_MAX_CNT	**The maximum number of RXD count in RX Ring #0**

10100918 **RX_CRX_IDX_1** **RX Ring #1 CPU pointer** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TX_CTX_IDX											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	TX_CTX_IDX	**Point to the next RXD CPU wants to use**

1010091C **RX_DRX_IDX_1** **RX Ring #1 DMA poitner** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					RX_DRX_IDX											
Type					RO											

Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	RX_DRX_IDX	**Point to the next RXD DMA wants to use**

10100A00 PDMA_INFO PDMA Information 1C000204

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					INDEX_WIDTH				BASE_PTR_WIDTH							
Type					RO				RO							
Reset					1	1	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_RING_NUM								TX_RING_NUM							
Type	RO								RO							
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

Bit(s)	Name	Description
27:24	INDEX_WIDTH	**Point to the next RXD CPU wants to use**
23:16	BASE_PTR_WIDTH	**Base pointer width, x** Base_addr[31:32-x] is shared with all ring base adderss. Only ring #0 base address[31:32-x] field Is writabl. [note]: "0" means no bit of base_address is shared.
15:8	RX_RING_NUM	**Rx ring number**
7:0	TX_RING_NUM	**Tx ring number**

10100A04 PDMA_GLO_CFG PDMA Global Configuration 00000050

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			HDR_SEG_LEN													
Type			RW													
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BIG_ENDIAN	TX_WB_DDONE	PDMA_BT_SIZE		RX_DMA_BUSY	RX_DMA_EN	TX_DMA_BUSY	TX_DMA_EN
Type									RW	RW	RW		RO	RW	RO	RW
Reset									0	1	0	1	0	0	0	0

Bit(s)	Name	Description
29:16	HDR_SEG_LEN	**Header Segment Length** Specify the header segment size in byte to supoprt RX header/payload scattering fucntion, when set to a non-zero value. When set to zero, the header/payload scattering feature is disabled.
7	BIG_ENDIAN	**Big endian** 0: PDMA will not do byte swapping for TX/RX packet header and payload 1: PDMA will do byte swaping for TX/RX packet header and payload
6	TX_WB_DDONE	0: Disable TX_DMA writing back DDONE into TXD 1: Enable TX_DMA writing back DDONE into TXD
5:4	PDMA_BT_SIZE	**The burst size of PDMA** 0: 4 DWORDs (16-bytes) 1: 8 DWORDs (32-bytes)

Bit(s)	Name	Description
		2: 16 DWORDs (64-bytes) 3: Reserved
3	RX_DMA_BUSY	0: RX_DMA is not busy 1: RX_DMA is busy
2	RX_DMA_EN	0: Diable RX_DMA (when disabled, RX_DMA will finish the current receiving packet, then stop) 1: Enable RX_DMA
1	TX_DMA_BUSY	0: TX_DMA is not busy 1: TX_DMA is busy
0	TX_DMA_EN	0: Disable TX_DMA (when disabled, TX_DMA will finish the current sending packet, then stop) 1: Enable TX_DMA

10100A0C **DELAY_INT_CFG** **Delay Interrupt Configuration** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TXDLY_INT_EN	TXMAX_PINT							TXMAX_PTIME							
Type	RW	RW							RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RXDLY_INT_EN	RXMAX_PINT							RXMAX_PTIME							
Type	RW	RW							RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	TXDLY_INT_EN	**Delay interrupt mechanism** 0: Disable TX delayed interrupt mechanism 1: Enable Tx delayed interrupt mechanism
30:24	TXMAX_PINT	**Specified Max. number of pended interrupts** When the number of pended interrupts is equal or greater than the value specified here or interupt pending time reach the limit (see below), an final TX_DLY_INT is generated. [Note] reset to 0 can disable pending interrupt count check.
23:16	TXMAX_PTIME	**Specified Max. pended time** When the pending time is equal or greater than TXMAX_PTIME x 20us or the number of pended TX_DONE is equal or greater than TXMAX_PINT 9see above), an final TX_DLY_INT is generated. [Note] reset to 0 can disable pending interrupt time check.
15	RXDLY_INT_EN	0: Disable Rx delayed interrupt mechanism 1: Enable Rx delayed interrupt mechanism
14:8	RXMAX_PINT	**Specified Max. number of pended interrupts** When the number of pended interrupts is equal or greater than the value specified here or interupt pending time reach the limit (see below), an final RX_DLY_INT is generated. [Note] reset to 0 can disable pending interrupt count check.
7:0	RXMAX_PTIME	**Specified Max. pended time** When the pending time is equal or greater than RXMAX_PTIME x 20us or the

Bit(s)	Name	Description
		number of pended RX_DONE is equal or greater than RXMAX_PINT 9see above), an finalRX_DLY_INT is generated. [Note] reset to 0 can disable pending interrupt time check.

10100A10 **FREEQ_THRES** **Free Queue Threshold** **00000002**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FREEQ_THRES			
Type													RW			
Reset													0	0	1	0

Bit(s)	Name	Description
3:0	FREEQ_THRES	**Rx free queue threshold** PDMA will stop DMA interface when left RX descriptors reach this threshold

10100A20 **INT_STATUS** **Interrupt Status** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_COHERENT	RX_DLY_INT	TX_COHERENT	TX_DLY_INT											RX_DONE_INT1	RX_DONE_INT0
Type	W1C	W1C	W1C	W1C											W1C	W1C
Reset	0	0	0	0											0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TX_DONE_INT3	TX_DONE_INT2	TX_DONE_INT1	TX_DONE_INT0
Type													W1C	W1C	W1C	W1C
Reset													0	0	0	0

Bit(s)	Name	Description
31	RX_COHERENT	**RX_DMA finds data coherent event while checking ddone bit.**
30	RX_DLY_INT	**Summary of the whole PDMA Rx related interrupts.**
29	TX_COHERENT	**TX_DMA finds data coherent event while checking ddone bit.**
28	TX_DLY_INT	**Summary of the whole PDMA Tx related interrupts.**
17	RX_DONE_INT1	**Rx ring #1 packet receive interrupt**
16	RX_DONE_INT0	**Rx ring #0 packet receive interrupt**
3	TX_DONE_INT3	**Tx ring #3 packet transmit interrupt**
2	TX_DONE_INT2	**Tx ring #2 packet transmit interrupt**
1	TX_DONE_INT1	**Tx ring #1 packet transmit interrupt**
0	TX_DONE_INT0	**Tx ring #0 packet transmit interrupt**

10100A28 **INT_MASK** **Interrupt Mask** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_COHERENT	RX_DLY_INT	TX_COHERENT	TX_DLY_INT											RX_DONE_INT1	RX_DONE_INT0
Type	RW	RW	RW	RW											RW	RW
Reset	0	0	0	0											0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TX_DONE_INT3	TX_DONE_INT2	TX_DONE_INT1	TX_DONE_INT0
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
31	RX_COHERENT	**Interrupt enable for RX_DMA data coherent vent** 0: Disable interrupt 1: Enable interrupt
30	RX_DLY_INT	**Summary of the whole PDMA Rx related interrupts.** 0: Disable interrupt 1: Enable interrupt
29	TX_COHERENT	**Interrupt enable for TX_DMA data coherent vent** 0: Disable interrupt 1: Enable interrupt
28	TX_DLY_INT	**Summary of the whole PDMA Tx related interrupts.** 0: Disable interrupt 1: Enable interrupt
17	RX_DONE_INT1	**Rx ring #1 packet receive interrupt** 0: Disable interrupt 1: Enable interrupt
16	RX_DONE_INT0	**Rx ring #0 packet receive interrupt** 0: Disable interrupt 1: Enable interrupt
3	TX_DONE_INT3	**Tx ring #3 packet transmit interrupt** 0: Disable interrupt 1: Enable interrupt
2	TX_DONE_INT2	**Tx ring #2 packet transmit interrupt** 0: Disable interrupt 1: Enable interrupt
1	TX_DONE_INT1	**Tx ring #1 packet transmit interrupt** 0: Disable interrupt 1: Enable interrupt
0	TX_DONE_INT0	**Tx ring #0 packet transmit interrupt** 0: Disable interrupt 1: Enable interrupt

10100A80 **PDMA_SCH** **Scheduler Configuration for Q0&Q1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SCH_MODE									
Type							RW									
Reset							0	0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

Bit(s)	Name	Description
25:24	SCH_MODE	**Scheduling Mode** 00: WRR 01: Strict priority, Q3>Q2,Q1>Q0 10: Mixed mode, Q3>WRR(Q2,Q1,Q0) 11: Mixed mode, Q3>Q2>WRR(Q1,Q0)

10100A84 PDMA_WRR Scheduler Configuration for Q2&Q3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		SCH_WT_Q3				SCH_WT_Q2				SCH_WT_Q1				SCH_WT_Q0		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Bit(s)	Name	Description
14:12	SCH_WT_Q3	**Scheduling Weight of TX Q3**
10:8	SCH_WT_Q2	**Scheduling Weight of TX Q2**
6:4	SCH_WT_Q1	**Scheduling Weight of TX Q1**
2:0	SCH_WT_Q0	**Scheduling Weight of TX Q0**

10100C00 SDM_CON Switch DMA Control 00078100

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REV0								PDMA_FC	PORT_MAP	LOOP_EN	TCO_81xx	UN_DROP_EN	UDPCS	TCPCS	IPCS
Type	RO								RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EXT_VLAN															
Type	RW															
Reset	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	REV0	**Reserved**
23	PDMA_FC	**TX PDMA Flow Cotnrol Enable** When this bit is set, the downstream fow control is enabled on PDMA 4 TX Ring (SDM_TRGING)

Bit(s)	Name	Description
		0: Disable 1: Enable
22	PORT_MAP	**RX Ring Selection** The received frame will be collected into the corresponding PDMA RX Ring based on the source port priority tag. 0: Priority Tag (SDMRRING[7:0]) 1: Source Port (SDM_RRING[12:8])
21	LOOP_EN	**Frame Engine Loop-back Mode Enable**
20	TCO_81xx	**Special tag Recongization Enable** When this bit is set, PDI(0x81xx) is recognized by the first byte (0x81) only. The second byte could be used for the specilqa purpose like the incoming source port.
19	UN_DROP_EN	**Drop Unknonwn MAC Address** 0: Disable 1: Enable
18	UDPCS	**UDP Packet Checksum RX Offload Enable** 0: disable, checksum result is showed on RX descriptor 1: enable, drop checksum error packet
17	TCPCS	**TCP Packet Checksum RX Offload Enable** 0: disable, checksum result is showed on RX descriptor 1: enable, drop checksum error packet
16	IPCS	**IP Header Checksum RX Offload Enable** 0: disable, checksum result is showed on RX descriptor 1: enable, drop checksum error packet
15:0	EXT_VLAN	**Outer VLAN Protocol ID** The specific vlaue is used to recognize the outer VLAN protocol ID only. Per inner VLAN or the general VLAN-tagged frame, the value PID=0x8100 is the uniqe protocol ID.

10100C04 **SDM_RING** **Switch DMA Rx Ring** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REV0												QUE3_RING_FC	QUE2_RING_FC	QUE1_RING_FC	QUE0_RING_FC
Type	RO												RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				PORT4_RING	PORT3_RING	PORT2_RING	PORT1_RING	PORT0_RING	PRI7_RING	PRI6_RING	PRI5_RING	PRI4_RING	PRI3_RING	PRI2_RING	PRI1_RING	PRI10_RING
Type				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:20	REV0	**Reserved**
19	QUE3_RING_FC	**Pause Switch Queue 3 by RX Ring##** When RX Ring# reaches the reserved free threshold(FREEQ_THRES), the queue 3 to CPU will be paused. 1: RX Ring #0 0: RX Ring #1

Bit(s)	Name	Description
18	QUE2_RING_FC	**Pause Switch Queue 2 by RX Ring##** When RX Ring# reaches the reserved free threshold(FREEQ_THRES), the queue 3 to CPU will be paused. 1: RX Ring #0 0: RX Ring #1
17	QUE1_RING_FC	**Pause Switch Queue 1 by RX Ring##** When RX Ring# reaches the reserved free threshold(FREEQ_THRES), the queue 3 to CPU will be paused. 1: RX Ring #0 0: RX Ring #1
16	QUE0_RING_FC	**Pause Switch Queue 0 by RX Ring##** When RX Ring# reaches the reserved free threshold(FREEQ_THRES), the queue 3 to CPU will be paused. 1: RX Ring #0 0: RX Ring #1
12	PORT4_RING	**Source Port 4 to RX Ring##** The received frames from the source port 4 will be sent to RX Ring# [Note] To use the source port, the special tag between FE and SW should be enabled. 1: RX Ring #0 0: RX Ring #1
11	PORT3_RING	**Source Port 3 to RX Ring##** The received frames from the source port 4 will be sent to RX Ring# [Note] To use the source port, the special tag between FE and SW should be enabled. 1: RX Ring #0 0: RX Ring #1
10	PORT2_RING	**Source Port 2 to RX Ring##** The received frames from the source port 4 will be sent to RX Ring# [Note] To use the source port, the special tag between FE and SW should be enabled. 1: RX Ring #0 0: RX Ring #1
9	PORT1_RING	**Source Port 1 to RX Ring##** The received frames from the source port 4 will be sent to RX Ring# [Note] To use the source port, the special tag between FE and SW should be enabled. 1: RX Ring #0 0: RX Ring #1
8	PORT0_RING	**Source Port 0 to RX Ring##** The received frames from the source port 4 will be sent to RX Ring# [Note] To use the source port, the special tag between FE and SW should be enabled. 1: RX Ring #0 0: RX Ring #1
7	PRI7_RING	**Priority 7 to RX Ring##** The received frames with priority tag 7 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1
6	PRI6_RING	**Priority 6 to RX Ring##** The received frames with priority tag 6 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1
5	PRI5_RING	**Priority 5 to RX Ring##** The received frames with priority tag 5 will be sent to RX Ring# 1: RX Ring #0

Bit(s)	Name	Description
		0: RX Ring #1
4	PRI4_RING	**Priority 4 to RX Ring##** The received frames with priority tag 4 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1
3	PRI3_RING	**Priority 3 to RX Ring##** The received frames with priority tag 3 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1
2	PRI2_RING	**Priority 2 to RX Ring##** The received frames with priority tag 2 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1
1	PRI1_RING	**Priority 1to RX Ring##** The received frames with priority tag 1 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1
0	PRI10_RING	**Priority 0 to RX Ring##** The received frames with priority tag 0 will be sent to RX Ring# 1: RX Ring #0 0: RX Ring #1

10100C08 SDM_TRING Switch DMA TX Ring **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RING3_WAN_FC				RING2_WAN_FC				RING1_WAN_FC				RING0_WAN_FC			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RING3_LAN_FC				RING2_LAN_FC				RING1_LAN_FC				RING0_LAN_FC			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:28	RING3_WAN_FC	**Pause TX Ring 3 by WAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: WAN port Queue#3 Bit.2: WAN port Queue#2 Bit.1: WAN port Queue#1 Bit.0: WAN port Queue#0
27:24	RING2_WAN_FC	**Pause TX Ring 2 by WAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: WAN port Queue#3 Bit.2: WAN port Queue#2 Bit.1: WAN port Queue#1 Bit.0: WAN port Queue#0
23:20	RING1_WAN_FC	**Pause TX Ring 1 by WAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: WAN port Queue#3 Bit.2: WAN port Queue#2 Bit.1: WAN port Queue#1

Bit(s)	Name	Description
		Bit.0: WAN port Queue#0
19:16	RING0_WAN_FC	**Pause TX Ring 0 by WAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: WAN port Queue#3 Bit.2: WAN port Queue#2 Bit.1: WAN port Queue#1 Bit.0: WAN port Queue#0
15:12	RING3_LAN_FC	**Pause TX Ring 3 by LAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: LAN port Queue#3 Bit.2: LAN port Queue#2 Bit.1: LAN port Queue#1 Bit.0: LAN port Queue#0
11:8	RING2_LAN_FC	**Pause TX Ring 2 by LAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: LAN port Queue#3 Bit.2: LAN port Queue#2 Bit.1: LAN port Queue#1 Bit.0: LAN port Queue#0
7:4	RING1_LAN_FC	**Pause TX Ring 1 by LAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: LAN port Queue#3 Bit.2: LAN port Queue#2 Bit.1: LAN port Queue#1 Bit.0: LAN port Queue#0
3:0	RING0_LAN_FC	**Pause TX Ring 0 by LAN Port** TX Ring# will be paused when the corresponding switch egress queue on WAN port is congested. Bit.3: LAN port Queue#3 Bit.2: LAN port Queue#2 Bit.1: LAN port Queue#1 Bit.0: LAN port Queue#0

10100C0C **SDM_MAC_ADRL** **Switch MAC Address LSB** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MAC_ADDR_LSB[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MAC_ADDR_LSB[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MAC_ADDR_LSB	**MAC Address bit[31:0]**

10100C10 **SDM_MAC_ADRH** **Switch MAC Address MSB** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MAC_ADDR_MSB															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	MAC_ADDR_MSB	**MAC Address bit[47:32]**

10100D00 **SDM_TPCNT** **Switch DMA Tx Packet Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TX_PCNT[31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TX_PCNT[15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_PCNT	**Transmit Packet Count**

10100D04 **SDM_TBCNT** **Switch DMA TX Byte Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TX_BCNT[31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TX_BCNT[15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	TX_BCNT	**Transmit Byte Count**

10100D08 **SDM_RPCNT** **Switch DMA RX Packet Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_PCNT[31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_PCNT[15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:0	RX_PCNT	**Receive Packet Count**

10100D0C **SDM_RBCNT** **Switch DMA RX Byte Count** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RX_BCNT[31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RX_BCNT[15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RX_BCNT	**Receive Byte Count**

10100D10 **SDM_CS_ERR** **Switch DMA RX Checksum Error** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CS_ERR_CNT[31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CS_ERR_CNT[15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CS_ERR_CNT	**Receive Checksum Error Count**

2.19 Switch Controller

2.19.1 Registers

ESW Changes LOG

Revision	Date	Author	Change Log
0.1	2013/5/29	PeterCT WU	Initialization

Module name: ESW Base address: (+10110000h)

Address	Name	Width	Register Function
10110000	ISR	32	Interrupt Status
10110004	IMR	32	Interrupt Mask
10110008	FCT0	32	Flow Control Threshold 0
1011000C	FCT1	32	Flow Control Threshold 1
10110010	PFC0	32	Priority Flow Control 0
10110014	PFC1	32	Priority Flow Control 1
10110018	PFC2	32	Priority Flow Control 2
1011001C	GQS0	32	Global Queue Status 0
10110020	GQS1	32	Global Queue Status 1
10110024	ATS	32	Address Table Search
10110028	ATS0	32	Address Table Status 0
1011002C	ATS1	32	Address Table Status 1
10110030	ATS2	32	Address Table Status 2
10110034	WMAD0	32	WT_MAC_AD0
10110038	WMAD1	32	WT_MAC_AD1
1011003C	WMAD2	32	WT_MAC_AD2
10110040	PVIDC0	32	PVID Configuration 0
10110044	PVIDC1	32	PVID Configuration 1
10110048	PVIDC2	32	PVID Configuration 2
1011004C	PVIDC3	32	PVID Configuration 3
10110050	VLANI0	32	VLAN Identifier 0
10110054	VLANI1	32	VLAN Identifier 1
10110058	VLANI2	32	VLAN Identifier 2
1011005C	VLANI3	32	VLAN Identifier 3
10110060	VLANI4	32	VLAN Identifier 4
10110064	VLANI5	32	VLAN Identifier 5
10110068	VLANI6	32	VLAN Identifier 6
1011006C	VLANI7	32	VLAN Identifier 7
10110070	VMSC0	32	VLAN Member Port Configuration 0
10110074	VMSC1	32	VLAN Member Port Configuration 1
10110078	VMSC2	32	VLAN Member Port Configuration 2
1011007C	VMSC3	32	VLAN Member Port Configuration 3
10110080	POA	32	Port Ability Offset
10110084	FPA	32	Force Port4 - Port0 Ability
10110088	PTS	32	Port Status
1011008C	SOCPC	32	SoC Port Control

10110090	POC0	32	Port Control 0
10110094	POC1	32	Port Control 1
10110098	POC2	32	Port Control 2
1011009C	SGC	32	Switch Global Control
101100A0	STRT	32	Switch Reset
101100A4	LEDP0	32	LED Port0
101100A8	LEDP1	32	LED Port1
101100AC	LEDP2	32	LED Port2
101100B0	LEDP3	32	LED Port3
101100B4	LEDP4	32	LED Port4
101100B8	WDTR	32	Watch Dog Trigger Reset
101100BC	DES	32	Debug Signal
101100C0	PCR0	32	PHY Control Register 0
101100C4	PCR1	32	PHY Control Register 1
101100C8	FPA1	32	Force P5P6 Ability
101100CC	FCT2	32	Flow Control Threshold 2
101100D0	QSS0	32	Queue Status 0
101100D4	QSS1	32	Queue Status 1
101100D8	DEC	32	Debug Control
101100DC	MTI	32	Memory Test Information
101100E0	PPC	32	Packet Counter
101100E4	SGC2	32	Switch Global Control 2
101100E8	P0PC	32	Port 0 Packet Counter
101100EC	P1PC	32	Port 1 Packet Counter
101100F0	P2PC	32	Port 2 Packet Counter
101100F4	P3PC	32	Port 3 Packet Counter
101100F8	P4PC	32	Port 4 Packet Counter
101100FC	P5PC	32	Port 5 Packet Counter
10110100	VUB0	32	VLAN Untag Block 0
10110104	VUB1	32	VLAN Untag Block 1
10110108	VUB2	32	VLAN Untag Block 2
1011010C	VUB3	32	VLAN Untag Block 3
10110110	BMU_CTRL	32	BC/MC/UN Rate Limit Control
10110114	BMU_LMT_NUM1	32	BC/MC/UN Rate Limit Frame Number
10110118	BMU_LMT_NUM2	32	BC/MC/UN Rate Limit Frame Number
1011011C	P01_ING_CTRL	32	Port 0&1 Ingress Rate Limit Control
10110120	P23_ING_CTRL	32	Port 2&3 Ingress Rate Limit Control
10110124	P45_ING_CTRL	32	Port 4&5 Ingress Rate Limit Control
10110128	P0_ING_THRES	32	Port 0 Ingress Rate Limit Threshold
1011012C	P1_ING_THRES	32	Port 1 Ingress Rate Limit Threshold
10110130	P2_ING_THRES	32	Port 2 Ingress Rate Limit Threshold
10110134	P3_ING_THRES	32	Port 3 Ingress Rate Limit Threshold
10110138	P4_ING_THRES	32	Port 4 Ingress Rate Limit Threshold

1011013C	P5_ING_THRES	32	**Port 5 Ingress Rate Limit Threshold**
10110140	P01_EG_CTRL	32	**Port 0/1 Egress Rate Limit Control**
10110144	P23_EG_CTRL	32	**Port 2/3 Egress Rate Limit Control**
10110148	P45_EG_CTRL	32	**Port 4/5 Egress Rate Limit Control**
1011014C	PCRI	32	**Packet Counter Recycle Indication**
10110150	P0TPC	32	**Port 0 TX Packet Counter**
10110154	P1TPC	32	**Port 1 TX Packet Counter**
10110158	P2TPC	32	**Port 2 TX Packet Counter**
1011015C	P3TPC	32	**Port 3 TX Packet Counter**
10110160	P4TPC	32	**Port 4 TX Packet Counter**
10110164	P5TPC	32	**Port 5 TX Packet Counter**
10110168	LEDC	32	**LED Control**

10110000 ISR Interrupt Status 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REV0		WATCHDOG1_TMR_EXPIRED	WATCHDOG0_TMR_EXPIRED	HAS_INTRUDER	PORT_ST_CHG	BC_STORM	MUST_DROP_LAN	GLOBAL_QUE_FULL			LAN_QUE_FULL_6	LAN_QUE_FULL_5	LAN_QUE_FULL_4	LAN_QUE_FULL_3	LAN_QUE_FULL_2
Type	RO		W1C	W1C	W1C	W1C	W1C	W1C	W1C			W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0			0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LAN_QUE_FULL_1	LAN_QUE_FULL_0														
Type	W1C	W1C														
Reset	0	0														

Bit(s)	Name	Description
31:30	REV0	**Reserved**
29	WATCHDOG1_TMR_EXPIRED	**P5 no transmit packet alert.** This bit indicating that P5 don't transmit packet for 3 seconds when P5 need to transmit packet. Write one clear. [Note] This feature is only valid when port 5 Giga MAC is implemented.
28	WATCHDOG0_TMR_EXPIRED	**Abnormal Alert** This bit indicating that global queue block counts is less than buf_starvation_th for 3 seconds. Write one clear.
27	HAS_INTRUDER	**Intruder Alert** This bit indicating that an unsecured packet is coming into a secured port. Write one clear.
26	PORT_ST_CHG	**Port status change** Any port from link status change. Write one clear.
25	BC_STORM	**BC storm**

Bit(s)	Name	Description
		The device is undergoing broadcast storm. Write one clear.
24	MUST_DROP_LAN	**Queue exhausted** The global queue is used up and all packets are dropped. Write one clear.
23	GLOBAL_QUE_FULL	**Global Queue Full.** Write one clear.
20	LAN_QUE_FULL_6	**Port 6 out queue full. Write one clear.** [Note]: This feature is only valid when port 5 Giga MAC is implemented.
19	LAN_QUE_FULL_5	**Port 5 out queue full. Write one clear.**
18	LAN_QUE_FULL_4	**Port 4 out queue full. Write one clear.**
17	LAN_QUE_FULL_3	**Port 3 out queue full. Write one clear.**
16	LAN_QUE_FULL_2	**Port 2 out queue full. Write one clear.**
15	LAN_QUE_FULL_1	**Port 1 out queue full. Write one clear.**
14	LAN_QUE_FULL_0	**Port 0 out queue full. Write one clear.**

10110004 **IMR** **Interrupt Mask** **FFFFFFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			WATCHDOG1_TMR_EXPIRED	WATCHDOG0_TMR_EXPIRED	HAS_INTRUDER	PORT_ST_CHG	BC_STORM	MUST_DROP_LAN	GLOBAL_QUE_FULL	REV1		LAN_QUE_FULL_6	LAN_QUE_FULL_5	LAN_QUE_FULL_4	LAN_QUE_FULL_3	LAN_QUE_FULL_2
Type			RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW
Reset			1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LAN_QUE_FULL_1	LAN_QUE_FULL_0														
Type	RW	RW														
Reset	1	1														

Bit(s)	Name	Description
29	WATCHDOG1_TMR_EXPIRED	**P5 no transmit packet alert.** This bit indicating that P5 don't transmit packet for 3 seconds when P5 need to transmit packet. Write one clear. [Note]: This feature is only valid when port 5 Giga MAC is implemented.
28	WATCHDOG0_TMR_EXPIRED	**Abnormal Alert** This bit indicating that global queue block counts is less than buf_starvation_th for 3 seconds. Write one clear.
27	HAS_INTRUDER	**Intruder Alert** This bit indicating that an unsecured packet is coming into a secured port. Write one clear.
26	PORT_ST_CHG	**Port status change** Any port from link status change. Write one clear.
25	BC_STORM	**BC storm** The device is undergoing broadcast storm. Write one clear.

Bit(s)	Name	Description
24	MUST_DROP_LAN	**Queue exhausted** The global queue is used up and all packets are dropped. Write one clear.
23	GLOBAL_QUE_FULL	**Global Queue Full.** Write one clear.
22:21	REV1	**Port 6 out queue full. Write one clear.** [Note]: This feature is only valid when port 5 Giga MAC is implemented.
20	LAN_QUE_FULL_6	**Port 6 out queue full. Write one clear.** [Note]: This feature is only valid when port 5 Giga MAC is implemented.
19	LAN_QUE_FULL_5	**Port 5 out queue full. Write one clear.**
18	LAN_QUE_FULL_4	**Port 4 out queue full. Write one clear.**
17	LAN_QUE_FULL_3	**Port 3 out queue full. Write one clear.**
16	LAN_QUE_FULL_2	**Port 2 out queue full. Write one clear.**
15	LAN_QUE_FULL_1	**Port 1 out queue full. Write one clear.**
14	LAN_QUE_FULL_0	**Port 0 out queue full. Write one clear.**

10110008 **FCT0** **Flow Control Threshold 0** **FFC86E5A**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FC_RLS_TH								FC_SET_TH							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DRO_RLS_TH								DROP_SET_TH							
Type	RW								RW							
Reset	0	1	1	0	1	1	1	0	0	1	0	1	1	0	1	0

Bit(s)	Name	Description
31:24	FC_RLS_TH	**Flow Control Release Threshold** Flow control will be disabled when the global queue block counts is greater than the release threshold
23:16	FC_SET_TH	**Flow Control Set Threshold** Flow control will be enabled when the global queue block counts is less than the set threshold
15:8	DRO_RLS_TH	**Drop Release Threshold** Switch will stop dropping packets when the global queue block counts is greater than the drop-release threshold
7:0	DROP_SET_TH	**Drop Set Threshold** Switch will start dropping packets when the global queue block counts is less than the drop-set threshold.

1011000C **FCT1** **Flow Control Threshold 1** **00000014**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									PORT_TH							
Type									RW							

Reset									0	0	0	1	0	1	0	0

Bit(s)	Name	Description
7:0	PORT_TH	**Per Port Output Threshold** When the global queue reaches the flow control or drop threshold on register FCT0, per port output threshold will be checked to enable flow-control or packet-dop depending on per queue minimum reserved blocks of the register PFC2.

10110010 PFC0 Priority Flow Control 0 0F000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					MTCC_LMT					TURN_OFF_FC						
Type					RW					RW						
Reset					1	1	1	1		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VO_NUM				CL_NUM				BE_NUM				BK_NUM			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
27:24	MTCC_LMT	**MTCC LIMIT** The maximum Back-off count limit to drop excessive collision packets.
22:16	TURN_OFF_FC	**Turn off FC When Receiving High Packet** Auto-turn-off FC when the programmed ports receive one of the highest priority packet. 0: DIsable 1: Enable
15:12	VO_NUM	**The proportional number of WRR for Voice Queue** After transmit exactly the number of packets then proceed to next queue. If equal to 0, only this queue is forced to the strict priority mode
11:8	CL_NUM	**The proportional number of WRR for Control-Load Queue** After transmit exactly the number of packet then proceed to next queue.
7:4	BE_NUM	**The proportional number of WRR for Best-Effort Queue** After transmit exactly the number of packet then proceed to next queue.
3:0	BK_NUM	**The proportional number of WRR for Background Queue After transmit exactly the number of packet then proceed to next queue.**

10110014 PFC1 Priority Flow Control 1 00001555

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CPU_USE_Q1_EN	EN_TOS							IGMP_TO_CPU	EN_VLAN						
Type	RW	RW							RW	RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PRIORITY_OP		PORT_PRI 6		PORT_PRI 5		PORT_PRI 4		PORT_PRI 3		PORT_PRI 2		PORT_PRI 1		PORT_PRI 0	

	TIO N															
Type	RW		RW		RW		RW		RW		RW		RW		RW	
Reset	0		0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31	CPU_USE_Q1_EN	**CPU Port only use q1 enable** 0: default priority resolution 1: packets forwarded to CPU port uses Best-Effor Queue
30:24	EN_TOS	**Port6 ~ port0 TOS_en.** Check TOS field of IP packets for priority resolution. [Note] Port 5 function is only vlaid when port 5 Giga MAC is implemented 0: Disable 1: Enable
23	IGMP_TO_CPU	**IGMP forward to CPU enable** 0: IGMP message will be floode to all ports 1: IGMP message will be forwarded to CPU port only.
22:16	EN_VLAN	**Enable per port VLAN-tag VID membership and priority tag check.** [Note] Port 5 function is only vlaid when port 5 Giga MAC is implemented 0: disable. 1: enable
15	PRIORITY_OPTION	**Priority Resolution Option** 0: 802.1p -> TOS -> Per port 1: TOS -> 802.1p -> Per port
13:12	PORT_PRI6	**Port priority** By setting this register to assign per port's default priority queue.
11:10	PORT_PRI5	**Port priority** By setting this register to assign per port's default priority queue. [Note] This feature is only valid when port 6 Giga MAC is implemented
9:8	PORT_PRI4	**Port priority** By setting this register to assign per port's default priority queue.
7:6	PORT_PRI3	**Port priority** By setting this register to assign per port's default priority queue.
5:4	PORT_PRI2	**Port priority** By setting this register to assign per port's default priority queue.
3:2	PORT_PRI1	**Port priority** By setting this register to assign per port's default priority queue.
1:0	PORT_PRI0	**Port priority** By setting this register to assign per port's default priority queue.

10110018 **PFC2** **Priority Flow Control 2** **03030303**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PRI_TH_VO								PRI_TH_CL							
Type	RW								RW							
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PRI_TH_BE								PRI_TH_BK							
Type	RW								RW							
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit(s)	Name	Description
31:24	PRI_TH_VO	**Voice Threshold (Highest Priority)**

Bit(s)	Name	Description
		The minimum reserved packet block count which outout queue can store when the flow-control /drop threshold of registers FTC0 and FCT1 is reached. If the queued blocks exceed the threshold, the incoming packet will be paused or dropped.
23:16	PRI_TH_CL	**Control Load Threshold** The minimum reserved packet block count which outout queue can store when the flow-control/drop threshold of registers FTC0 and FCT1 is reached. If the queued blocks exceed the threshold, the incoming packet will be paused or dropped.
15:8	PRI_TH_BE	**Best Effort threshold** The minimum reserved packet block count which outout queue can store when the flow-control/drop threshold of registers FTC0 and FCT1 is reached. If the queued blocks exceed the threshold, the incoming packet will be paused or dropped.
7:0	PRI_TH_BK	**Background Threshold (Lowest Priority)** The minimum reserved packet block count which outout queue can store when the flow-control/drop threshold of registers FTC0 and FCT1 is reached. If the queued blocks exceed the threshold, the incoming packet will be paused or dropped.

1011001C **GQS0** **Global Queue Status 0** **FA41016E**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PRI7_QUE**		**PRI6_QUE**		**PRI5_QUE**		**PRI4_QUE**		**PRI3_QUE**		**PRI2_QUE**		**PRI1_QUE**		**PRI0_QUE**	
Type	RW		RW		RW		RW		RW		RW		RW		RW	
Reset	1	1	1	1	1	0	1	0	0	1	0	0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**EMPTY_CNT**								
Type								RO								
Reset								1	0	1	1	0	1	1	1	0

Bit(s)	Name	Description
31:30	PRI7_QUE	**Queue mapping for Priority Tag #7**
29:28	PRI6_QUE	**Queue mapping for Priority Tag #6**
27:26	PRI5_QUE	**Queue mapping for Priority Tag #5**
25:24	PRI4_QUE	**Queue mapping for Priority Tag #4**
23:22	PRI3_QUE	**Queue mapping for Priority Tag #3**
21:20	PRI2_QUE	**Queue mapping for Priority Tag #2**
19:18	PRI1_QUE	**Queue mapping for Priority Tag #1**
17:16	PRI0_QUE	**Queue mapping for Priority Tag #0**
8:0	EMPTY_CNT	**Global Queue Block Counts** This field indicates the number of block count left in the global free queue.

10110020 **GQS1** **Global Queue Status 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**OUTQUE_FULL_VO**								**OUTQUE_FULL_CL**							
Type	RO								RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	OUTQUE_FULL_BE								OUTQUE_FULL_BK							
Type	RO								RO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:24	OUTQUE_FULL_VO	**Congested Voice Queue** The corresponding queue is congested
23:16	OUTQUE_FULL_CL	**Congested Control Load Queue** The corresponding queue is congested
15:8	OUTQUE_FULL_BE	**Congested Best Effort Queue** The corresponding queue is congested
7:0	OUTQUE_FULL_BK	**Congested Background Queue** The corresponding queue is congested

10110024 ATS Address Table Search 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name														AT_LKUP_IDLE	SEARCH_NXT_ADDR	BEGIN_SEARCH_ADDR
Type														RO	W1C	W1C
Reset														1	0	0

Bit(s)	Name	Description
2	AT_LKUP_IDLE	**Address Lookup Idle** This field indicates that Adress Table engine is in IDLE state.
1	SEARCH_NXT_ADDR	**Search For The Next Address (Self_Clear)**
0	BEGIN_SEARCH_ADDR	**Start Searching The Address Table (Self_Clear)**

10110028 ATS0 Address Table Status 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HASH_ADD_LU													R_PORT_MAP[6:4]		
Type	RO													RO		
Reset	0	0	0	0	0	0	0	0	0	0				0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	R_PORT_MAP[3:0]					R_VLD				R_AGE_FIELD				R_MC_INGRESS	AT_TABLE_END	SEARCH_RDY

Type	RO					RO				RO				RO	RO	RC
Reset	0	0	0	0		0	0	0	0	0	0	0		0	0	0

Bit(s)	Name	Description
31:22	HASH_ADD_LU	**Address table lookup address**
18:12	R_PORT_MAP	**Port map** The MAC existing in the bit =1.
10:7	R_VLD	**VLAN index**
6:4	R_AGE_FIELD	**Aging field**
2	R_MC_INGRESS	**MC Ingress**
1	AT_TABLE_END	**Search to the end of address table**
0	SEARCH_RDY	**Data is ready (read clear)**

1011002C **ATS1** **Address Table Status 1** **0000000**
0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MAC_AD_SER0**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	MAC_AD_SER0	**Read MAC Address [15:0]**

10110030 **ATS2** **Address Table Status 2** **0000000**
0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**MAC_AD_SER0[31:16]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**MAC_AD_SER0[15:0]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	MAC_AD_SER0	**Read MAC Address [31:16]**

10110034 **WMAD0** **WT_MAC_AD0** **0008000**
0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**HASH_ADD_LU**												**AT_CFG_IDLE**	**W_PORT_MAP[6:4]**		
Type	RO												RO	RW		

Reset	0	0	0	0	0	0	0	0	0	0			1	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**W_PORT_MAP[3:0]**					**W_INDEX**				**W_AGE_FIELD**			**SA_FILTER**	**W_MC_INGRESS**	**W_MAC_DONE**	**W_MAC_CMD**
Type	RW					RW				RW			RW	RW	RO	WO
Reset	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:22	HASH_ADD_LU	**Address table configuration address**
19	AT_CFG_IDLE	**Address Table Configuration SM IDLE**
18:12	W_PORT_MAP	**Write Port map**
10:7	W_INDEX	**VLAN index** 0: VLAN 0 1-14: ... 15: VLAN 15
6:4	W_AGE_FIELD	**Write Aging field** 3'b111: static address, 3'b001 - 3'b110: the entry is valid and will be aged out 2'b000: default, entry is invalid
3	SA_FILTER	**SA_FILTER** 0: default 1: The corresponding packet will be dropped when the SA is matched
2	W_MC_INGRESS	**Write MC Ingress**
1	W_MAC_DONE	**MAC Write Done** 0: default 1: MAC address write OK (read clear)
0	W_MAC_CMD	**MAC Address write Command** 0: default 1: the MAC write data is ready and write to MAC table now(self_clear)

10110038 **WMAD1** **WT_MAC_AD1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**W_MAC_15_0**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	W_MAC_15_0	**Write MAC Address[15:0]**

1011003C **WMAD2** **WT_MAC_AD2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**W_MAC_47_16[31:16]**															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**W_MAC_47_16[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	W_MAC_47_16	**Write MAC Address[47:16]**

10110040 **PVIDC0** **PVID Configuration 0** **00001001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**P1_PVID[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**P1_PVID[3:0]**				**P0_PVID**											
Type	RW				RW											
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
23:12	P1_PVID	**Port1 PVID Setting**
11:0	P0_PVID	**Port0 PVID Setting**

10110044 **PVIDC1** **PVID Configuration 1** **00001001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**P3_PVID[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**P3_PVID[3:0]**				**P2_PVID**											
Type	RW				RW											
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
23:12	P3_PVID	**Port3 PVID Setting**
11:0	P2_PVID	**Port2 PVID Setting**

10110048 **PVIDC2** **PVID Configuration 2** **00001001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**P5_PVID[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**P5_PVID[3:0]**				**P4_PVID**											
Type	RW				RW											
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description

Bit(s)	Name	Description
23:12	P5_PVID	**Port5 PVID Setting** [Note] This feature is only valid when port 5 Giga MAC is implemented.
11:0	P4_PVID	**Port4 PVID Setting**

1011004C **PVIDC3** **PVID Configuration 3** **75020001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		QUE3_PRIT				QUE2_PRIT				QUE1_PRIT				QUE0_PRIT		
Type		RW				RW				RW				RW		
Reset		1	1	1		1	0	1		0	0	0		0	1	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					P6_PVID											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
30:28	QUE3_PRIT	**Priority Tag Egress Mapping for Voice Queue#3**
26:24	QUE2_PRIT	**Priority Tag Egress Mapping for Control Load Queue#2**
22:20	QUE1_PRIT	**Priority Tag Egress Mapping for Best Effort Queue#1**
18:16	QUE0_PRIT	**Priority Tag Egress Mapping for Back Ground Queue#0**
11:0	P6_PVID	**Port6 PVID Setting**

10110050 **VLANI0** **VLAN Identifier 0** **00002001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									VID1[11:4]							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	VID1[3:0]				VID0											
Type	RW				RW											
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
23:12	VID1	**VLAN Field Identifier for VLAN 1**
11:0	VID0	**VLAN Field Identifier for VLAN 0**

10110054 **VLANI1** **VLAN Identifier 1** **00004003**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									VID3[11:4]							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	VID3[3:0]				VID2											
Type	RW				RW											
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit(s)	Name	Description
23:12	VID3	**VLAN Field Identifier for VLAN 3**
11:0	VID2	**VLAN Field Identifier for VLAN 2**

10110058 **VLANI2** **VLAN Identifier 2** **0000600**
5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**VID5[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VID5[3:0]**				**VID4**											
Type	RW				RW											
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit(s)	Name	Description
23:12	VID5	**VLAN Field Identifier for VLAN 5**
11:0	VID4	**VLAN Field Identifier for VLAN 4**

1011005C **VLANI3** **VLAN Identifier 3** **0000800**
7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**VID7[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VID7[3:0]**				**VID6**											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit(s)	Name	Description
23:12	VID7	**VLAN Field Identifier for VLAN 7**
11:0	VID6	**VLAN Field Identifier for VLAN 6**

10110060 **VLANI4** **VLAN Identifier 4** **0000A00**
9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**VID9[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VID9[3:0]**				**VID8**											
Type	RW				RW											
Reset	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit(s)	Name	Description
23:12	VID9	**VLAN Field Identifier for VLAN 9**
11:0	VID8	**VLAN Field Identifier for VLAN 8**

10110064 **VLANI5** **VLAN Identifier 5** **0000C00B**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**VID11[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VID11[3:0]**				**VID10**											
Type	RW				RW											
Reset	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Bit(s)	Name	Description
23:12	VID11	**VLAN Field Identifier for VLAN 11**
11:0	VID10	**VLAN Field Identifier for VLAN 10**

10110068 **VLANI6** **VLAN Identifier 6** **0000E00D**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**VID13[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VID13[3:0]**				**VID12**											
Type	RW				RW											
Reset	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit(s)	Name	Description
23:12	VID13	**VLAN Field Identifier for VLAN 13**
11:0	VID12	**VLAN Field Identifier for VLAN 12**

1011006C **VLANI7** **VLAN Identifier 7** **0001000F**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									**VID15[11:4]**							
Type									RW							
Reset									0	0	0	0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**VID15[3:0]**				**VID14**											
Type	RW				RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit(s)	Name	Description
23:12	VID15	**VLAN Field Identifier for VLAN 15**
11:0	VID14	**VLAN Field Identifier for VLAN 14**

10110070 **VMSC0** **VLAN Member Port Configuration 0** **FFFFFFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**VLAN_MEMSET_3**								**VLAN_MEMSET_2**							

Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_MEMSET_1								VLAN_MEMSET_0							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:24	VLAN_MEMSET_3	**VLAN 3 Member Port**
23:16	VLAN_MEMSET_2	**VLAN 2 Member Port**
15:8	VLAN_MEMSET_1	**VLAN 1 Member Port**
7:0	VLAN_MEMSET_0	**VLAN 0 Member Port**

10110074 **VMSC1** **VLAN Member Port Configuration 1** **FFFFFFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VLAN_MEMSET_7								VLAN_MEMSET_6							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_MEMSET_5								VLAN_MEMSET_4							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:24	VLAN_MEMSET_7	**VLAN 7 Member Port**
23:16	VLAN_MEMSET_6	**VLAN 6 Member Port**
15:8	VLAN_MEMSET_5	**VLAN 5 Member Port**
7:0	VLAN_MEMSET_4	**VLAN 4 Member Port**

10110078 **VMSC2** **VLAN Member Port Configuration 2** **FFFFFFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VLAN_MEMSET_11								VLAN_MEMSET_10							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_MEMSET_9								VLAN_MEMSET_8							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:24	VLAN_MEMSET_11	**VLAN 11 Member Port**
23:16	VLAN_MEMSET_10	**VLAN 10 Member Port**
15:8	VLAN_MEMSET_9	**VLAN 9 Member Port**
7:0	VLAN_MEMSET_8	**VLAN 8 Member Port**

1011007C **VMSC3** **VLAN Member Port Configuration 3** **FFFFFFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VLAN_MEMSET_15								VLAN_MEMSET_14							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_MEMSET_13								VLAN_MEMSET_12							
Type	RW								RW							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:24	VLAN_MEMSET_15	**VLAN 15 Member Port**
23:16	VLAN_MEMSET_14	**VLAN 14 Member Port**
15:8	VLAN_MEMSET_13	**VLAN 13 Member Port**
7:0	VLAN_MEMSET_12	**VLAN 12 Member Port**

10110080 **POA** **Port Ability Offset** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	G1_LINK	G0_LINK	LINK					G1_TXC		G0_TXC		XFC				
Type	RO	RO	RO					RO		RO		RO				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DUPLEX							G1_SPD		G0_SPD		SPEED				
Type	RO							RO		RO		RO				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	G1_LINK	**Port 6 Link Status** 1: Link up 0: Link down
30	G0_LINK	**Port 5 Link Status** [Note] This feature is only valid when port 5 giga MAC is implemented. 1: Link up 0: Link down
29:25	LINK	**Port 4 ~ port0 Link Status** 1: Link up 0: Link down
24:23	G1_TXC	**Flow Control Status fo Port6** The flow control capability status bit after Auto-negotiation or force mode. 1xb: full duplex and tx flow control ON x1b: full duplex and rx flow control ON 00b: flow control off
22:21	G0_TXC	**Flow Control Status fo Port5** The flow control capability status bit after Auto-negotiation or force mode. [Note] This feature is only valid when port 5 giga MAC is implemented. 1xb: full duplex and tx flow control ON x1b: full duplex and rx flow control ON 00b: flow control off
20:16	XFC	**Flow Control Status of port 0 ~ 4** The flow control capability status bit after Auto-negotiation or force mode. 0: flow control off 1: full duplex and 802.3x flow control ON (after AN or forced)
15:9	DUPLEX	**Port6 ~ port0 Duplex Mode**

Bit(s)	Name	Description
		[Note]: Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: half duplex 1: full duplex
8:7	G1_SPD	**MII port 6 Speed:Mode** 10: 1000M 01: 100M 00: 10M
6:5	G0_SPD	**MII port 5 Speed:Mode** [Note] This feature is only valid when port 5 Giga MAC is implemented 10: 1000M 01: 100M 00: 10M
4:0	SPEED	**Port4 ~ port0 Speed Mode** 0: 10M 1: 100M

10110084 **FPA** **Force Port4 - Port0 Ability** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FORCE_MODE					FORCE_LINK						FORCE_XFC				
Type	RW					RW						RW				
Reset	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				FORCE_DPX							XTAL_COMP	FORCE_SPD				
Type				RW							RW	RW				
Reset				0	0	0	0	0			0	0	0	0	0	0

Bit(s)	Name	Description
31:27	FORCE_MODE	**Port4 ~ port 0 force mode** 0: default 1: force mode. Auto-negotiation status is ignored. All the port ability are forced according to the following fields of the register FPA.
26:22	FORCE_LINK	**Port 4 ~ port 0 PHY Link** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 1: Link up 0: Link down
20:16	FORCE_XFC	**Port 4 ~ port 0 Flow control of PHY port** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 0: default OFF 1: 802.3x flow control ON
12:8	FORCE_DPX	**Flow Control Status of port 0 ~ 4** The flow control capability status bit after Auto-negotiation or force mode. 0: flow control off 1: full duplex and 802.3x flow control ON (after AN or forced)
5	XTAL_COMP	**Crystal rate compensation** 0: Disable 1: When the switch has transmitted 20000 bytes, the switch will compensate for the loss of crystal rate.
4:0	FORCE_SPD	**Port4 ~ port0 Speed:**

Bit(s)	Name	Description
		This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 1: 100M 0: 10M

10110088 **PTS** **Port Status** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							G1_TXC_STATUS	G0_TXC_STATUS		SECURED_ST						
Type							RO	RO		RO						
Reset							0	0		0	0	0	0	0	0	0

Bit(s)	Name	Description
9	G1_TXC_STATUS	**Port 6 TXC status** 0: no alert 1: error, no TXC
8	G0_TXC_STATUS	**Port 5 TXC status** [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: no alert 1: error, no TXC
6:0	SECURED_ST	**Security Status** 0: no alert 1: has intruder coming if turn on the SA_secured mode, read clear

1011008C **SOCPC** **SoC Port Control** **027F7F7F**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							CRC_PADDING	CPU_SELECTION		DISBC2CPU						
Type							RW	RW		RW						
Reset							1	0	0	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		DISMC2CPU								DISUN2CPU						
Type		RW								RW						
Reset		1	1	1	1	1	1	1		1	1	1	1	1	1	1

Bit(s)	Name	Description
25	CRC_PADDING	**CRC padding from CPU** If this bit is set , all packets from CPU don't need to append CRC and the outgoing LAN/WAN port will calculate and append CRC. 0: packets from CPU need CRC appending 1: packets from CPU without CRC appending

Bit(s)	Name	Description
24:23	CPU_SELECTION	**CPU Selection** 00b: Port 6 01b: Port 0 10b: Port 4 11b: Port 5
22:16	DISBC2CPU	**Disable BC to CPU** When this bit = 1, BC frames from the corresponding port will not be forward to CPU. [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: Includes CPU port. 1: Excludes CPU port
14:8	DISMC2CPU	**Disable MC to CPU** When this bit =1, MC frames from the corresponding port will not forward to CPU. [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: Includes CPU port. 1: Excludes CPU port
6:0	DISUN2CPU	**Disable UN to CPU** When this bit =1, Unkonwn frames from the corresponding port will not forward to CPU. [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: Includes CPU port. 1: Excludes CPU port

10110090 **POC0** **Port Control 0** **3F807F7F**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**HASH_ADDR_SHIFT**		**DIS_GMII_PORT_1**	**DIS_GMII_PORT_0**	**DIS_PORT**					**DISRMC2_CPU**						
Type	RW		RW	RW	RW					RW						
Reset	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**EN_FC**							**MAC_FCP_OPTION**	**EN_BP**						
Type		RW							RW	RW						
Reset		1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:30	HASH_ADDR_SHIFT	**Address table hashing algorithm option for member set index**
29	DIS_GMII_PORT_1	**Disable port 6** 0: port enable 1: port disable
28	DIS_GMII_PORT_0	**Disable port 5** [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: port enable 1: port disable
27:23	DIS_PORT	**Disable phy port**

Bit(s)	Name	Description
		0: port enable 1: port disable
22:16	DISRMC2_CPU	**Unknown Reserved Multicast Frame Excludes CPU** [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: Unknown Reserved Multicast Forward Rule (SGC.RMC_RULE) 1: Excludes CPU port
14:8	EN_FC	**Apply 802.3x status after Auto-negotiation** This field can individually control the 802.3x capability after Auto-negotiation is done. [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: ignore the AN stats for 802.3x capability 1: follow the AN status for 802.3x capability
7	MAC_FCP_OPTION	**Multicast Flow control/Backpressure option** 0: When all ports are fc/bp disable, the switch will use drop_threshold to drop frames only. If not, the switch will use fc_threshold and drop_threshold. 1: When only the destination TX port is fc/bp disable, the switch will use drop_threshold to drop frames only . If not, that TX port uses fc_threshold and drop_threshold.
6:0	EN_BP	**Apply back pressure capability** [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: ignore the back pressure mode (default OFF) 1: apply back pressure based on SGC.BP_MODE.

10110094 **POC1** **Port Control 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			DISIPMC2CPU							BLOCKING_STATE						
Type			RW							RW						
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		DIS_LRNING								SA_SECURED_PORT						
Type		RW								RW						
Reset		0	0	0	0	0	0	0		0	0	0	0	0	0	0

Bit(s)	Name	Description
29:23	DISIPMC2CPU	**Unknown IP Multicast Frame Excludes CPU** 0: Unknown IP Multicast Forward Rule (SGC.IP_MULT_RULE) 1: Excludes CPU port
22:16	BLOCKING_STATE	**Port State for Spanning Tree Protocol** [Note]: Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: normal state 1: blocking state, forwarding rmc packet to cpu(need programming address table
14:8	DIS_LRNING	**Disable SA learning** [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: default enabled 1: disable Source MAC learning
6:0	SA_SECURED_PORT	**SA secured mode** [Note*1]: Must set dis_learn and sa_secured at the same time. [Note*2] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: don't care SA match, 1: the packets' SA needs match, otherwise discard the packets

10110098	POC2	Port Control 2	00007F00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		G1_TXC_CHECK	G0_TXC_CHECK				MLD2CPU_EN	IPV6_MULT_RULE		DIS_UC_PAUSE						
Type		RW	RW				RW	RW		RW						
Reset		0	0				0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PER_VLAN_UNTAG_EN	ENAGING_PORT								UNTAG_EN						
Type	RW	RW								RW						
Reset	0	1	1	1	1	1	1	1		0	0	0	0	0	0	0

Bit(s)	Name	Description
30	G1_TXC_CHECK	**Check the port 6 TXC** if no txc clock, then disable MII port 0: disable 1: enable, check TXC
29	G0_TXC_CHECK	**Check the port 5 TXC** if no txc clock, then disable MII port [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: disable 1: enable, check TXC
25	MLD2CPU_EN	**MLD Message Packets forward to CPU** 0: MLD message will be flooded 1: MLD message will be forward to CPU port only
24:23	IPV6_MULT_RULE	**Unknown IPV6 Multicast Frame Forward Rule** If no match in the address table, then folloing the rule 00: BC 01: to CPU 10: drop 11: Reserved
22:16	DIS_UC_PAUSE	**Disable Unicast Pause Frame** [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: switch will consider pause frame when DA!=0180c20001 but unicast to CPU, 1: switch will not consider pause frame when DA!= 0180c20001 and unicast to CPU
15	PER_VLAN_UNTAG_EN	**Per port per vlan untag enable** VLAN tag removal option. 0: Use per port UNTAG_EN 1: Use untag enable bitmap in VLAN table
14:8	ENAGING_PORT	**Port aging** [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: disable aging that the MAC address is belong to programmed port(s) 1: enable aging
6:0	UNTAG_EN	**Per Port VLAN Tag Temoval** [Note] Port5 funciton is only valid when port 5 Giga MAC is implemented. 0: disable

Bit(s)	Name	Description
		1: enable VLAN tag field removal.

1011009C **SGC** **Switch Global Control** **6008A041**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		BKOFF_ALG	LEN_ERR_CHK	IP_MULT_RULE		RMC_RULE		LED_FLASH_TIME		BISH_TH		BISH_DIS	BP_MODE		DISMIIPORT_WASTX	
Type		RW	RW	RW		RW		RW		RW		RW	RW		RW	
Reset		1	1	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BP_JAM_CNT				DISABLE_TX_BACKOFF	ADDRESS_HASH_ALG		DIS_PKT_TX_ABORT	PKT_MAX_LEN		BC_STORM_PROT		AGING_INTERNAL			
Type	RW				RW	RW		RW	RW		RW		RW			
Reset	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit(s)	Name	Description
30	BKOFF_ALG	**Backoff Algorithm Option** 0: default 1: comply with UNH test
29	LEN_ERR_CHK	**Length of Received Frame Check Enable** When the bit is set, the received packet length will be checked for length encapsulated frames. 0: default disabled 1: comply with UNH test
28:27	IP_MULT_RULE	**Unknown IP Multicase Frame Forward Rule** If no match in the address table, then following the rules. 00: BC 01: to cpu 10: drop 11: reserved
26:25	RMC_RULE	**Unknown Reserved Multicast Frame Forward Rule** If no match in the address table, then follow the rules. 00: to all port(not include blocking state port) 01: to cpu 10: drop 11: reserved
24:23	LED_FLASH_TIME	**The Frequency Of LED Flash** 00: 30ms 01: 60ms 10: 240ms 11: 480ms
22:21	BISH_TH	**The Threshold Of Memory Bisshop** 11: skip if fail 8 blocks, 0 00: skip if fail 16 (default, from pins) 01: skip if fail 48 10: skip if fail 64
20	BISH_DIS	**Build In Self Hop**

Bit(s)	Name	Description
		0: enable skip function 1: disable
19:18	BP_MODE	**Back Pressure Mode** 00: disable 01: BP jam, the jam number is set by bp_num 10: BP jamALL, jam packet until the BP condition is released(default), 11: BP carrier, use carrier insertion to do back pressure
17:16	DISMIIPORT_WASTX	**GMII Port Disable Was_Transmit** [Note] This feature is only valid when port 5 Giga MAC is implemented. 1: disable was_transmit (good for late CRS PHY, like HPNA2.0 or power-LAN), 0: enable
15:12	BP_JAM_CNT	**Back Pressure Jam Number** The consecutive jam count when back pressure is enabled, The default is 10 packet jam then one no-jam packet.
11	DISABLE_TX_BACKOFF	**Disable The Collision Back Off Timer** 0: default 1: re-transmit immediately after collision
10:9	ADDRESSS_HASH_ALG	**MAC Address Hashing Algorithm** 00: direct mode, using last 10-bit as hashing address 01: XOR48 mode 10: XOR32 mode 11: reserved
8	DIS_PKT_TX_ABORT	**Disable Packet TX Abort** 1: Disable collision 16 packet abort and late collision abort 0: enable both abort
7:6	PKT_MAX_LEN	**Maximum Packet Length** Untaged / VLAN-taged 00: 1536 Bytes / 1536 Bytes 01: 1518 Bytes / 1522 Bytes 10: 1522 Bytes / 1526 Bytes 11: Reserved / Reserved
5:4	BC_STORM_PROT	**Global Broadcast Storm Protection** BC will be blocked, if the following number of BC blocks in in output queues 00: disable 01: 64 10: 96 11: 128
3:0	AGING_INTERNAL	**Aging Timer** 0000: disable age 0001: 300sec 0010 - 0111: 600 ~ 38400sec 1xxx: Fast Age (60sec)

101100A0 **STRT** **Switch Reset** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESET_SW[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESET_SW[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	RESET_SW	**Reset switch engine, data, address, link memory , cpu port and ahb interface when writing data to the STRT register.**

101100A4 **LEDP0** **LED Port0** **00000005**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**P0_LED**			
Type													RW			
Reset													0	1	0	1

Bit(s)	Name	Description
3:0	P0_LED	**port0 LED state, default = link/activity** 0000: link 0001: 100M speed 0010: duplex 0011: activity 0100: collision 0101: link/activity 0110: duplex/collsion 0111: 10M speed/activity 1000: 100M speed/activity 1011: off 1100: on 1010: blink

101100A8 **LEDP1** **LED Port1** **00000005**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**P1_LED**			
Type													RW			
Reset													0	1	0	1

Bit(s)	Name	Description
3:0	P1_LED	**port1 LED state, default = link/activity**

101100AC **LEDP2** **LED Port2** **00000005**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name													P2_LED			
Type													RW			
Reset													0	1	0	1

Bit(s)	Name	Description
3:0	P2_LED	**port2 LED state, default = link/activity**

101100B0 **LEDP3** **LED Port3** **00000005**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													P3_LED			
Type													RW			
Reset													0	1	0	1

Bit(s)	Name	Description
3:0	P3_LED	**port3 LED state, default = link/activity**

101100B4 **LEDP4** **LED Port4** **00000005**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													P4_LED			
Type													RW			
Reset													0	1	0	1

Bit(s)	Name	Description
3:0	P4_LED	**port4 LED state, default = link/activity**

101100B8 **WDTR** **Watch Dog Trigger Reset** **0000001E**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUF_STARV_TH							
Type									RW							
Reset									0	0	0	1	1	1	1	0

Bit(s)	Name	Description
7:0	BUF_STARV_TH	**Buffer starvation threshold** Switch will interrupt CPU when the global queue block counts is less than the threshold for 3 seconds.

101100BC **DES** **Debug Signal** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DEBUG_SIGNAL**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
15:0	DEBUG_SIGNAL	**Port 5 Debug Signal**

101100C0 **PCR0** **PHY Control Register 0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**WT_NWAY_DATA**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESV0**	**RD_PHY_CMD**	**WT_PHY_CMD**	**CPU_PHY_REG**								**CPU_PHY_ADDR**				
Type	RO	RW	RW	RW								RW				
Reset	0	0	0	0	0	0	0	0				0	0	0	0	0

Bit(s)	Name	Description
31:16	WT_NWAY_DATA	**The Data Be Written into PHY**
15	RESV0	**Reserved**
14	RD_PHY_CMD	**Read command** To enable read command on PHY, write 1 to this bit . After command is completed, this bit is self-cleared.
13	WT_PHY_CMD	**Write command** To enable write command on PHY, write 1 to this bit . After command is completed, this bit is self-cleared
12:8	CPU_PHY_REG	**PHY register address**
4:0	CPU_PHY_ADDR	**PHY address** (Note: The internal 5-ports PHY reserves the PHY address starting from 5'd0 ~ 5'd4. For the external PHY, the PHY address from 5'd5 to 5'd31 can be applied. The default PHY address of Port 5 is 5'd5 for auto-polling function.)

101100C4 **PCR1** **PHY Control Register 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RD_DATA**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**RD_R**	**WT_D**

Bit																
Name															DY	ONE
Type															RC	RC
Reset															0	0

Bit(s)	Name	Description
31:16	RD_DATA	**The Read Data**
1	RD_RDY	**Read Operation is Done**
0	WT_DONE	**Write Operation is Done**

101100C8 **FPA1** **Force P5P6 Ability** **05500328**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			AP_EN	EXT_PHY_ADDR_BASE					G0_RXCLK_SKEW_SEL		G0_TXCLK_SKEW_SEL			TURBO_MII_CLK		
Type			RW	RW					RW		RW			RW		
Reset			0	0	0	1	0	1	0	1	0	1		0		
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name			FORCE_RGMII_LINK1	FORCE_RGMII_LINK0	FORCE_RGMII_EN1	FORCE_RGMII_EN0	FORCE_RGMII_XFC1		FORCE_RGMII_XFC0		FORCE_RGMII_DPX1	FORCE_RGMII_DPX0	FORCE_RGMII_SPD1		FORCE_RGMII_SPD0	
Type			RW	RW	RW	RW	RW		RW		RW	RW	RW		RW	
Reset			0	0	0	0	1	1	0	0	1	0	1	0	0	0

Bit(s)	Name	Description
29	AP_EN	**Port 5 Auto Polling Enable** [Note] This feature is only valid when port 5 Giga MAC is implemented.
28:24	EXT_PHY_ADDR_BASE	**Port 5 External PHY Base Address** [Note] This feature is only valid when port 5 Giga MAC is implemented.
23:22	G0_RXCLK_SKEW_SEL	**Port 5 RXCLK Skew Selection** [Note] This feature is only valid when port 5 Giga MAC is implemented.
21:20	G0_TXCLK_SKEW_SEL	**Port 5 TXCLK Skew Selection** [Note] This feature is only valid when port 5 Giga MAC is implemented.
18	TURBO_MII_CLK	**Port 5 revMII Mode Clock Selection** [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: 25MHz output clock 1: 31.25MHz output clock
13	FORCE_RGMII_LINK1	**Force Port 6 Link** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 0: link down 1: link up
12	FORCE_RGMII_LINK0	**Force Port 5 Link** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: link down 1: link up

Bit(s)	Name	Description
11	FORCE_RGMII_EN1	**Force Port 6 Enable** 0: reserved 1: force mode. Auto-negotiation status is ignored. Port 5 ability is forced according to the following fields of the register FPA1.
10	FORCE_RGMII_EN0	**Force Port 5 Enable** [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: default 1: force mode. Auto-negotiation status is ignored. Port 5 ability is forced according to the following fields of the register FPA1.
9:8	FORCE_RGMII_XFC1	**Force port 6 flow control ability** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 1x: for tx x1: for rx
7:6	FORCE_RGMII_XFC0	**Force port 5 flow control ability** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. [Note] This feature is only valid when port 5 Giga MAC is implemented. 1x: for tx x1: for rx
5	FORCE_RGMII_DPX1	**Force port 6 duplex** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 0: half duplex 1: full duplex
4	FORCE_RGMII_DPX0	**Force port 5 duplex** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. [Note] This feature is only valid when port 5 Giga MAC is implemented. 0: half duplex 1: full duplex
3:2	FORCE_RGMII_SPD1	**Force port 6 speed** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. 1x: 1GbpsMhz 01: 100MbpsMHz 00: 10MbpsMHz
1:0	FORCE_RGMII_SPD0	**Force port 5 speed** This field is valid only FORCE_MDOE is set. The final resolution is reported to POA register. [Note] This feature is only valid when port 5 Giga MAC is implemented. 1x: 1GbpsMhz 01: 100MbpsMHz 00: 10MbpsMHz

101100CC **FCT2** **Flow Control Threshold 2** **0000A30C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								DIS_IPV6MC2CPU							MUST_DROP_RLS_TH[4:3]	
Type								RW							RW	
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MUST_DROP_RL			MUST_DROP_SET_TH							MC_PER_PORT_TH					

	S_TH[2:0]															
Type	RW			RW							RW					
Reset	1	0	1	0	0	0	1	1			0	0	1	1	0	0

Bit(s)	Name	Description
24:18	DIS_IPV6MC2CPU	**Unknown IPv6 Multicast Frame Excludes CPU** 0: Unknown IPv6 Multicast Forward Rule (POC2.IPV6_MULT_RULE) 1: Exclude CPU port
17:13	MUST_DROP_RLS_TH	**If the global queue pointer higher than the threshold. The must drop condition will be released.**
12:8	MUST_DROP_SET_TH	**If the global queue pointer reach msut drop threshold. All incoming packets have to be dropped.**
5:0	MC_PER_PORT_TH	**MC packets per port threshold.** When the global queue reaches the flow control threshold on register FCT0, per port output threshold for MC packet will be checked to enable flow-control or packet-drop on imncoming MC packets.

101100D0 QSS0 Queue Status 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									BE_CNT_R[8:1]							
Type									RO							
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BE_CNT_R[0:0]	BK_CNT_R										SEE_CNT_PORT_SEL				
Type	RO	RO										RW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
23:15	BE_CNT_R	**Link control best effort queue block counter monitor.**
14:5	BK_CNT_R	**Link control background queue block counter monitor.**
4:0	SEE_CNT_PORT_SEL	**Link control bock couonter port selection**

101100D4 QSS1 Queue Status 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															VO_CNT_R[8:7]	
Type															RO	
Reset															0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VO_CNT_R[6:0]							CL_CNT_R								
Type	RO							RO								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
17:9	VO_CNT_R	**Link control voice queue block counter monitor.**
8:0	CL_CNT_R	**Link control control queue block counter monitor.**

101100D8 **DEC** **Debug Control** **40400100**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SW2FE_IPG								FE2SW_IPG							
Type	RW								RW							
Reset	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								BRIDGE_EN								
Type								RW								
Reset								1								

Bit(s)	Name	Description
31:24	SW2FE_IPG	**SW2FE Bridge IPG Byte Count** Inter-Frame Byte Count between the consecutive frames flowing from Switch to Frame Engine
23:16	FE2SW_IPG	**FE2SW Bridge IPG byte count** Inter-Frame Byte Count between the consecurtive frames flowing from Frame Engine to Switch
8	BRIDGE_EN	**Enable FE2SW Bridge IPG Prevention** 1'b0: Disable 1'b1: Enable IPG Prevention when FE2SW_BRIDGE_IPG is too short (8'd16) to receive the next frame.

101100DC **MTI** **Memory Test Information** **0000006A**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										SW_RAM_TEST_DONE	LK_RAM_TEST_DONE	LK_RAM_TEST_FAIL	AT_RAM_TEST_DONE	AT_RAM_TEST_FAIL	DT_RAM_TEST_DONE	DT_RAM_TEST_FAIL
Type										RO	RO	RO	RO	RO	RO	RO
Reset										1	1	0	1	0	1	0

Bit(s)	Name	Description
6	SW_RAM_TEST_DONE	**Switch Memory Ram Test Done**
5	LK_RAM_TEST_DONE	**Link Ram Test Done**
4	LK_RAM_TEST_FAIL	**Link Ram Test Fail**
3	AT_RAM_TEST_DONE	**Address Table Ram Test Done**
2	AT_RAM_TEST_FAI	**Address Table Ram Test Fail**

Bit(s)	Name	Description
	L	
1	DT_RAM_TEST_DONE	**Data Buffer Ram Test Done**
0	DT_RAM_TEST_FAIL	**Data Buffer Ram Test Fail**

101100E0 PPC Packet Counter 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SW2FE_CNT**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FE2SW_CNT**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	SW2FE_CNT	**SW2FE_CNT Switch to frame engine packet counter**
15:0	FE2SW_CNT	**FE2SW_CNT Frame engine to switch packet counter**

101100E4 SGC2 Switch Global Control 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**P6_RXFC_QUE_EN**	**P6_TXFC_WL_EN**	**LAN_PMAP**						**SPECIAL_TAG_EN**	**TX_CPU_TPID_BIT_MAP**						
Type	RW	RW	RW						RW	RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**P6_TXFC_QUE_EN**	**ARBITER_LAN_EN**	**CPU_TPID_EN**	**ARBITER_GPT_EN**	**SLOT_4TO1**		**DOUBLE_TAG_EN**						
Type				RW	RW	RW	RW	RW		RW						
Reset				0	0	0	0	0		0	0	0	0	0	0	0

Bit(s)	Name	Description
31	P6_RXFC_QUE_EN	**Port 6 RX flow control on per egress queue** 0: Port 6 RX flow control will pause all 4 egress queue 1: Port 6 RX flow control will pause 4 egress queue independently according to the corresponding congestion signals.
30	P6_TXFC_WL_EN	**Port 6 TX flow controll by Switch WAN/LAN port** 0: Port 6 TX flow control is decided by any port and any queue of the Switch congestion 1: Port 6 TX flow control is decided by WAN/LAN port of the Switch congestion separately.
29:24	LAN_PMAP	**Lan port bit map**

Bit(s)	Name	Description
		This field indicates per port attribute used for flow control. (Note: Port5 funciton is only valid when port 5 Giga MAC is implemented) 1: Lan port 0: Wan port
23	SPECIAL_TAG_EN	**Special Tag enable** 0: default; RX special tag is enabled according to the global control bit-CPU_TPID_EN. TX special tag is enabled according to the per-port TX_CPU_TPID_BIT_MAP 1: CPU_TPID_EN is not used Both TX and RX special tag feature are decided by the per-port TX_CPU_TPID_BIT_MAP
22:16	TX_CPU_TPID_BIT_MAP	**Transmit CPU TPID(810x) port bit map** 0: default (TPID=0x8100) 1: TPID=0x810? depending on TX/RX usages (Note: Port5 funciton is only valid when port 5 Giga MAC is implemented)
12	P6_TXFC_QUE_EN	**Port 6 per queue TX flow control** This bit is only valid when P6_TXFC_WL_EN is enabled. 0: 4 congest signals to Frame Engine are decided by the wired-or result of all egress queues on Switch WAN/LAN ports. 1: 4 congest signals to Frame Engine are decided by the individual and the corresponding 4 egress queues on Switch WAN/LAN ports.
11	ARBITER_LAN_EN	**Memory arbiter only for P0~P4 enable** 0: default 1: memory arbiter only for P0~P4.
10	CPU_TPID_EN	**CPU TPID(81xx) enable** 0: disable. CPU TPID=8100 1: enable. CPU TPID=810x.
9	ARBITER_GPT_EN	**Memory Arbiter only for P5 and P6** 0: default 1: Enable
8	SLOT_4TO1	**Memory Arbiter Ratio Selection** 0: (P5,P6) : (P0-P4) = 3:2 1: (P5,P6) : (P0-P4) = 4:1
6:0	DOUBLE_TAG_EN	**Insert double tag field** When this bit is set , the incoming packet is allowed to insert outer or double tag. 1: enable double tag field 0: disable the double tag field. (Note: Port5 funciton is only valid when port 5 Giga MAC is implemented)

101100E8 **P0PC** **Port 0 Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT0	**Port 0 Receive Bad Packet Counter**
15:0	GOOD_PKT_CNT0	**Port 0 Receive Good Packet Counter**

101100EC **P1PC** **Port 1 Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT1															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT1															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT1	**Port 1 Receive Bad Packet Counter**
15:0	GOOD_PKT_CNT1	**Port 1 Receive Good Packet Counter**

101100F0 **P2PC** **Port 2 Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT2															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT2															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT2	**Port 2 Receive Bad Packet Counter**
15:0	GOOD_PKT_CNT2	**Port 2 Receive Good Packet Counter**

101100F4 **P3PC** **Port 3 Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT3															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT3															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT3	**Port 3 Receive Bad Packet Counter**
15:0	GOOD_PKT_CNT3	**Port 3 Receive Good Packet Counter**

101100F8 **P4PC** **Port 4 Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT4															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT4															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT4	**Port 4 Receive Bad Packet Counter**
15:0	GOOD_PKT_CNT4	**Port 4 Receive Good Packet Counter**

101100FC **P5PC** **Port 5 Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT5															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT5															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT5	**Port 5 Receive Bad Packet Counter**
15:0	GOOD_PKT_CNT5	**Port 5 Receive Good Packet Counter**

10110100 **VUB0** **VLAN Untag Block 0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					VLAN_3_UNTAG_EN							VLAN_2_UNTAG_EN[6:2]				
Type					RW							RW				
Reset					0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_2_UNTAG_EN[1:0]		VLAN_1_UNTAG_EN							VLAN_0_UNTAG_EN						
Type	RW		RW							RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
27:21	VLAN_3_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 3**
20:14	VLAN_2_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 2**
13:7	VLAN_1_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 1**
6:0	VLAN_0_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 0**

10110104 VUB1 VLAN Untag Block 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					VLAN_7_UNTAG_EN							VLAN_6_UNTAG_EN[6:2]				
Type					RW							RW				
Reset					0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_6_UNTAG_EN[1:0]		VLAN_5_UNTAG_EN							VLAN_4_UNTAG_EN						
Type	RW		RW							RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
27:21	VLAN_7_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 7**
20:14	VLAN_6_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 6**
13:7	VLAN_5_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 5**
6:0	VLAN_4_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 4**

10110108 VUB2 VLAN Untag Block 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					VLAN_11_UNTAG_EN							VLAN_10_UNTAG_EN[6:2]				
Type					RW							RW				
Reset					0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VLAN_10_UNTAG_EN[1:0]		VLAN_9_UNTAG_EN							VLAN_8_UNTAG_EN						
Type	RW		RW							RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
27:21	VLAN_11_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 11**
20:14	VLAN_10_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 10**
13:7	VLAN_9_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 9**
6:0	VLAN_8_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 8**

1011010C VUB3 VLAN Untag Block 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					VLAN_15_UNTAG_EN							VLAN_14_UNTAG_EN[6:2]				
Type					RW							RW				
Reset					0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	VLAN_14_UNTAG_EN[1:0]		VLAN_13_UNTAG_EN							VLAN_12_UNTAG_EN						
Type	RW		RW							RW						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
27:21	VLAN_15_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 15**
20:14	VLAN_14_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 14**
13:7	VLAN_13_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 13**
6:0	VLAN_12_UNTAG_EN	**Port 0 ~ 6 untag_en of VLAN 12**

10110110 **BMU_CTRL** **BC/MC/UN Rate Limit Control** **7C000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		ONE_US_CYCLE_NUM								P5_RATE_LIMIT_CTRL				P4_RATE_LIMIT_CTRL		
Type		RW								RW				RW		
Reset		1	1	1	1	1	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		P3_RATE_LIMIT_CTRL				P2_RATE_LIMIT_CTRL				P1_RATE_LIMIT_CTRL				P0_RATE_LIMIT_CTRL		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Bit(s)	Name	Description
30:24	ONE_US_CYCLE_NUM	**One micro-second Cycle Number** This field is used to calculate 1us period
22:20	P5_RATE_LIMIT_CTRL	**Port 5 rate Limit Control** (Note: This feature is only alid when port 5 GMAC is implemented)
18:16	P4_RATE_LIMIT_CTRL	**Port 4 rate Limit Control**
14:12	P3_RATE_LIMIT_CTRL	**Port 3 rate Limit Control**
10:8	P2_RATE_LIMIT_CTRL	**Port 2 rate Limit Control**
6:4	P1_RATE_LIMIT_CTRL	**Port 1 rate Limit Control**
2:0	P0_RATE_LIMIT_CTRL	**Port 0 rate Limit Control** 2: Broadcast frame enable 1: Multicast frame enable 0: Unknown frame enable

10110114 **BMU_LMT_NUM1** **BC/MC/UN Rate Limit Frame Number** **FFFFFFFF**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RATE_LIMIT_NUM_100M															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RATE_LIMIT_NUM_10M															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31:16	RATE_LIMIT_NUM_100M	**Rate Limit Received BC/MC/UN frame number in 100M in 100ms duration**
15:0	RATE_LIMIT_NUM_10M	**Rate Limit Received BC/MC/UN frame number in 10M in 1s duration**

10110118 BMU_LMT_NUM2 BC/MC/UN Rate Limit Frame Number 1818FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IG_RATE_BYTE_OPT	IG_RATE_BYTE_NUM							EG_RATE_BYTE_OPT	EG_RATE_BYTE_NUM						
Type	RW	RW							RW	RW						
Reset	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RATE_LIMIT_NUM_1000M															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
31	IG_RATE_BYTE_OPT	**Ingress Rate Byte Option** 0: Add 1: Minus
30:24	IG_RATE_BYTE_NUM	**Ingress Rate Byte Number**
23	EG_RATE_BYTE_OPT	**Egress Rate Byte Option** 0: Add 1: Minus
22:16	EG_RATE_BYTE_NUM	**Egress Rate Byte Number**
15:0	RATE_LIMIT_NUM_1000M	**Rate Limit Received BC/MC/UN frame number in 1000M in 10ms duration** (note: This feature is only valid whe port 5 GMAC is implemented))

1011011C P01_ING_CTRL Port 0&1 Ingress Rate Limit Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		P1_INGRESS_CTRL	P1_MNG_PKT_BYPASS	P1_INGRESS_FLOW_CTRL_ON	P1_TIMER_TICK		P1_TOKEN									

Type		RW	RW	RW	RW		RW									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**P0_INGRESS_CTRL**	**P0_MNG_PKT_BYPASS**	**P0_INGRESS_FLOW_CTRL_ON**	**P0_TIMER_TICK**		**P0_TOKEN**									
Type		RW	RW	RW	RW		RW									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
30	P1_INGRESS_CTRL	**Port1 Ingress Limit Control** 0: OFF 1: ON
29	P1_MNG_PKT_BYPASS	**Port1 Management Packet ByPass** 0: All packet included 1: Mangement Frame Excluded
28	P1_INGRESS_FLOW_CTRL_ON	**Port 1 Ingress rate Flow Control** When the bit is set, the pause frame is used prior to packet dropped according to P1_ING_THRES. If the bucket is empty, then P1 will start to discard the received packets except those specific packet in P1_MNG_PKY_BYPASS mode. 0: OFF 1: ON
27:26	P1_TIMER_TICK	**Port 1 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
25:16	P1_TOKEN	**Port 1 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes
14	P0_INGRESS_CTRL	**Port 0 Ingress Limit Control** 0: OFF 1: ON
13	P0_MNG_PKT_BYPASS	**Port 0 Management Packet ByPass** 0: All packet included 1: Mangement Frame Excluded
12	P0_INGRESS_FLOW_CTRL_ON	**Port 0 Ingress rate Flow Control** When the bit is set, the pause frame is used prior to packet dropped according to P0_ING_THRES. If the bucket is empty, then P0 will start to discard the received packets except those specific packet in P0_MNG_PKY_BYPASS mode. 0: OFF 1: ON
11:10	P0_TIMER_TICK	**Port 0 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
9:0	P0_TOKEN	**Port 0 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte)

Bit(s)	Name	Description
		The maximum space of this bucket is 16'hFFFF bytes

10110120	P23_ING_CTRL	Port 2&3 Ingress Rate Limit Control	00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		P3_INGRESS_CTRL	P3_MNG_PKT_BYPASS	P3_INGRESS_FLOW_CTRL_ON	P3_TIMER_TICK		P3_TOKEN									
Type		RW	RW	RW	RW		RW									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		P2_INGRESS_CTRL	P2_MNG_PKT_BYPASS	P2_INGRESS_FLOW_CTRL_ON	P2_TIMER_TICK		P2_TOKEN									
Type		RW	RW	RW	RW		RW									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
30	P3_INGRESS_CTRL	**Port 3 Ingress Limit Control** 0: OFF 1: ON
29	P3_MNG_PKT_BYPASS	**Port 3 Management Packet ByPass** 0: All packet included 1: Mangement Frame Excluded
28	P3_INGRESS_FLOW_CTRL_ON	**Port 3 Ingress rate Flow Control** When the bit is set, the pause frame is used prior to packet dropped according to P3_ING_THRES. If the bucket is empty, then P3 will start to discard the received packets except those specific packet in P3 MNG_PKY_BYPASS mode. 0: OFF 1: ON
27:26	P3_TIMER_TICK	**Port 3 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
25:16	P3_TOKEN	**Port 3 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes
14	P2_INGRESS_CTRL	**Port 2 Ingress Limit Control** 0: OFF 1: ON
13	P2_MNG_PKT_BYPASS	**Port 2 Management Packet ByPass** 0: All packet included

Bit(s)	Name	Description
		1: Mangement Frame Excluded
12	P2_INGRESS_FLOW_CTRL_ON	**Port 2 Ingress rate Flow Control** When the bit is set, the pause frame is used prior to packet dropped according to P2_ING_THRES. If the bucket is empty, then P2 will start to discard the received packets except those specific packet in P2_MNG_PKY_BYPASS mode. 0: OFF 1: ON
11:10	P2_TIMER_TICK	**Port 2 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
9:0	P2_TOKEN	**Port 2 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes

10110124 **P45_ING_CTRL** **Port 4&5 Ingress Rate Limit Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		P5_INGRESS_CTRL	P5_MNG_PKT_BYPASS	P5_INGRESS_FLOW_CTRL_ON	P5_TIMER_TICK		P5_TOKEN									
Type		RW	RW	RW	RW		RW									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		P4_INGRESS_CTRL	P4_MNG_PKT_BYPASS	P4_INGRESS_FLOW_CTRL_ON	P4_TIMER_TICK		P4_TOKEN									
Type		RW	RW	RW	RW		RW									
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
30	P5_INGRESS_CTRL	**Port 5 Ingress Limit Control** 0: OFF 1: ON
29	P5_MNG_PKT_BYPASS	**Port 5 Management Packet ByPass** 0: All packet included 1: Mangement Frame Excluded
28	P5_INGRESS_FLOW_CTRL_ON	**Port 5 Ingress rate Flow Control** When the bit is set, the pause frame is used prior to packet dropped according to P5_ING_THRES. If the bucket is empty, then P5 will start to discard the received packets except those specific packet in P5_MNG_PKY_BYPASS mode.

Bit(s)	Name	Description
		0: OFF 1: ON
27:26	P5_TIMER_TICK	**Port 5 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
25:16	P5_TOKEN	**Port 5 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes
14	P4_INGRESS_CTRL	**Port 4 Ingress Limit Control** 0: OFF 1: ON
13	P4_MNG_PKT_BYPASS	**Port 4 Management Packet ByPass** 0: All packet included 1: Mangement Frame Excluded
12	P4_INGRESS_FLOW_CTRL_ON	**Port 4 Ingress rate Flow Control** When the bit is set, the pause frame is used prior to packet dropped according to P4_ING_THRES. If the bucket is empty, then P4 will start to discard the received packets except those specific packet in P4_MNG_PKY_BYPASS mode. 0: OFF 1: ON
11:10	P4_TIMER_TICK	**Port 4 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
9:0	P4_TOKEN	**Port 4 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes

10110128 **P0_ING_THRES** **Port 0 Ingress Rate Limit Threshold** **AAAA5555**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P0_IN_FCOFF_THRES															
Type	RW															
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P0_IN_FCON_THRES															
Type	RW															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31:16	P0_IN_FCOFF_THRES	**Port 0 ingress rate limit flow control off.** If P0_INGRESS_FLOW_CTRL_ON = 1 and P0 Flow control capability is on (XFC status in 0x80), then P0 will initiate PAUSE OFF frame or stop backpressure.
15:0	P0_IN_FCON_THRES	**Port 0 ingress rate limit flow control on.** If P0_INGRESS_FLOW_CTRL_ON = 1 and P0 Flow control capability is on (XFC status in 0x80), then P0 will initiate PAUSE ON frame or backpressure.

1011012C	P1_ING_THRES	Port 1 Ingress Rate Limit Threshold	AAAA5555

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P1_IN_FCOFF_THRES															
Type	RW															
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P1_IN_FCON_THRES															
Type	RW															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31:16	P1_IN_FCOFF_THRES	**Port 1 ingress rate limit flow control off.** If P1_INGRESS_FLOW_CTRL_ON = 1 and P1 Flow control capability is on (XFC status in 0x80), then P2 will initiate PAUSE OFF frame or stop backpressure.
15:0	P1_IN_FCON_THRES	**Port 1 ingress rate limit flow control on.** If P1_INGRESS_FLOW_CTRL_ON = 1 and P1 Flow control capability is on (XFC status in 0x80), then P1 will initiate PAUSE ON frame or backpressure.

10110130	P2_ING_THRES	Port 2 Ingress Rate Limit Threshold	AAAA5555

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P2_IN_FCOFF_THRES															
Type	RW															
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P2_IN_FCON_THRES															
Type	RW															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31:16	P2_IN_FCOFF_THRES	**Port 2 ingress rate limit flow control off.** If P2_INGRESS_FLOW_CTRL_ON = 1 and P2 Flow control capability is on (XFC status in 0x80), then P2 will initiate PAUSE OFF frame or stop backpressure.
15:0	P2_IN_FCON_THRES	**Port 2 ingress rate limit flow control on.** If P2_INGRESS_FLOW_CTRL_ON = 1 and P2 Flow control capability is on (XFC status in 0x80), then P2 will initiate PAUSE ON frame or backpressure.

10110134	P3_ING_THRES	Port 3 Ingress Rate Limit Threshold	AAAA5555

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P3_IN_FCOFF_THRES															
Type	RW															
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P3_IN_FCON_THRES															
Type	RW															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31:16	P3_IN_FCOFF_THRES	**Port 3 ingress rate limit flow control off.** If P3_INGRESS_FLOW_CTRL_ON = 1 and P3 Flow control capability is on (XFC status in 0x80), then P3 will initiate PAUSE OFF frame or stop backpressure.
15:0	P3_IN_FCON_THRES	**Port 3 ingress rate limit flow control on.** If P3_INGRESS_FLOW_CTRL_ON = 1 and P3 Flow control capability is on (XFC status in 0x80), then P3 will initiate PAUSE ON frame or backpressure.

10110138 **P4_ING_THRES** **Port 4 Ingress Rate Limit Threshold** **AAAA5555**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P4_IN_FCOFF_THRES															
Type	RW															
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P4_IN_FCON_THRES															
Type	RW															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31:16	P4_IN_FCOFF_THRES	**Port 4 ingress rate limit flow control off.** If P4_INGRESS_FLOW_CTRL_ON = 1 and P4 Flow control capability is on (XFC status in 0x80), then P4 will initiate PAUSE OFF frame or stop backpressure.
15:0	P4_IN_FCON_THRES	**Port 4 ingress rate limit flow control on.** If P4_INGRESS_FLOW_CTRL_ON = 1 and P4 Flow control capability is on (XFC status in 0x80), then P4 will initiate PAUSE ON frame or backpressure.

1011013C **P5_ING_THRES** **Port 5 Ingress Rate Limit Threshold** **AAAA5555**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	P5_IN_FCOFF_THRES															
Type	RW															
Reset	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	P5_IN_FCON_THRES															
Type	RW															
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit(s)	Name	Description
31:16	P5_IN_FCOFF_THRES	**Port 5 ingress rate limit flow control off.** If P5_INGRESS_FLOW_CTRL_ON = 1 and P5 Flow control capability is on (XFC status in 0x80), then P5 will initiate PAUSE OFF frame or stop backpressure. (note: This feature is only valid when port 5 Giga MAC is implemented)
15:0	P5_IN_FCON_THRES	**Port 5 ingress rate limit flow control on.** If P5_INGRESS_FLOW_CTRL_ON = 1 and P0 Flow control capability is on (XFC status in 0x80), then P5 will initiate PAUSE ON frame or backpressure. (note: This feature is only valid when port 5 Giga MAC is implemented)

10110140 **P01_EG_CTRL** **Port 0/1 Egress Rate Limit Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				P1_EGRESS_CTRL	P1_TIMER_TICK		P1_TOKEN									
Type				RW	RW		RW									
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				P0_EGRESS_CTRL	P0_TIMER_TICK		P0_TOKEN									
Type				RW	RW		RW									
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
28	P1_EGRESS_CTRL	**Port 1 Egress Control** 1: ON 0: OFF
27:26	P1_TIMER_TICK	**Port 1 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
25:16	P1_TOKEN	**Port 1 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes 1: ON 0: OFF
12	P0_EGRESS_CTRL	**Port 0 Egress Control** 0: 512us 1: 128us 2: 32us 3: 8us
11:10	P0_TIMER_TICK	**Port 0 Timer Tick** 1: ON 0: OFF
9:0	P0_TOKEN	**Port 0 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes

10110144 **P23_EG_CTRL** **Port 2/3 Egress Rate Limit Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				P3_EGRESS_CT	P3_TIMER_TICK		P3_TOKEN									

				RL												
Type				RW	RW		RW									
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**P2_EGRESS_CTRL**	**P2_TIMER_TICK**		**P2_TOKEN**									
Type				RW	RW		RW									
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
28	P3_EGRESS_CTRL	**Port 3 Egress Control** 1: ON 0: OFF
27:26	P3_TIMER_TICK	**Port 3 Timer Tick** 0: 512us 1: 128us 2: 32us 3: 8us
25:16	P3_TOKEN	**Port 3 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes 1: ON 0: OFF
12	P2_EGRESS_CTRL	**Port 2 Egress Control** 0: 512us 1: 128us 2: 32us 3: 8us
11:10	P2_TIMER_TICK	**Port 2 Timer Tick** 1: ON 0: OFF
9:0	P2_TOKEN	**Port 2 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes

10110148 **P45_EG_CTRL** **Port 4/5 Egress Rate Limit Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				**P5_EGRESS_CTRL**	**P5_TIMER_TICK**		**P5_TOKEN**									
Type				RW	RW		RW									
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**P4_EGRESS_CTRL**	**P4_TIMER_TICK**		**P4_TOKEN**									

Type				RW	RW		RW									
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
28	P5_EGRESS_CTRL	**Port 5 Egress Control** (Note: This feature is only valid when port 5 Giga MAC is implemented) 1: ON 0: OFF
27:26	P5_TIMER_TICK	**Port 5 Timer Tick** (Note: This feature is only valid when port 5 Giga MAC is implemented) 0: 512us 1: 128us 2: 32us 3: 8us
25:16	P5_TOKEN	**Port 5 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes (Note: This feature is only valid when port 5 Giga MAC is implemented) 1: ON 0: OFF
12	P4_EGRESS_CTRL	**Port 4 Egress Control** 0: 512us 1: 128us 2: 32us 3: 8us
11:10	P4_TIMER_TICK	**Port 4 Timer Tick** 1: ON 0: OFF
9:0	P4_TOKEN	**Port 4 Token** Every timer tick, Token number bytes will be added into the bucket. (Unit : Byte) The maximum space of this bucket is 16'hFFFF bytes

1011014C **PCRI** **Packet Counter Recycle Indication** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PKT_CNT_CLR**		**TCOL_PKT_REC**							**TXOK_PKT_REC**						
Type	WO		RO							RO						
Reset	0		0	0	0	0	0	0		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**BAD_PKT_REC**							**GOOD_PKT_REC**						
Type			RO							RO						
Reset			0	0	0	0	0	0		0	0	0	0	0	0	0

Bit(s)	Name	Description
31	PKT_CNT_CLR	**Tx/Rx Packet Counters Write One Clear** When this bit is set,all Tx/Rx packet counters will be clear.This bit can be self-clear automatically.
29:24	TCOL_PKT_REC	**Per Port Transmitted Collision Packet Counter Recycle** This bit indicates that the per port transmitted collision packet counter recycles the count. Write one clear.

Bit(s)	Name	Description
22:16	TXOK_PKT_REC	**Per Port Transmitted Good Packet Counter Recycle** This bit indicates that the per port transmitted good packet counter recycles the count. Write one clear.
13:8	BAD_PKT_REC	**Per Port Received Bad Packet Counter Recycle** This bit indicates that the per port received badd packet counter recycles the count. Write one clear.
6:0	GOOD_PKT_REC	**Per Port Received Good Packet Counter Recycle** This bit indicates that the per port received good packet counter recycles the count. Write one clear.

10110150 **P0TPC** **Port 0 TX Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT0															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT0	**Port 0 packet counte for transmitted packets with collisionautomatically.**
15:0	GOOD_PKT_CNT0	**Port 0 packet counter for transmitted packets successfully**

10110154 **P1TPC** **Port 1 TX Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT1															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT1															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT1	**Port 1 packet counte for transmitted packets with collisionautomatically.**
15:0	GOOD_PKT_CNT1	**Port 1 packet counter for transmitted packets successfully**

10110158 **P2TPC** **Port 2 TX Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT2															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GOOD_PKT_CNT2															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT2	**Port 2 packet counte for transmitted packets with collisionautomatically.**
15:0	GOOD_PKT_CNT2	**Port 2 packet counter for transmitted packets successfully**

1011015C **P3TPC** **Port 3 TX Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT3															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT3															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT3	**Port 3 packet counte for transmitted packets with collisionautomatically.**
15:0	GOOD_PKT_CNT3	**Port 3 packet counter for transmitted packets successfully**

10110160 **P4TPC** **Port 4 TX Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT4															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT4															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT4	**Port 4 packet counte for transmitted packets with collisionautomatically.**
15:0	GOOD_PKT_CNT4	**Port 4 packet counter for transmitted packets successfully**

10110164 **P5TPC** **Port 5 TX Packet Counter** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BAD_PKT_CNT5															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOOD_PKT_CNT5															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:16	BAD_PKT_CNT5	**Port 5 packet counte for transmitted packets with collisionautomatically.** (Note: This feature is only valid when port 5 Giga MAC is implemented)
15:0	GOOD_PKT_CNT5	**Port 5 packet counter for transmitted packets successfully** (Note: This feature is only valid when port 5 Giga MAC is implemented)

10110168 **LEDC** **LED Control** **00E00000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**OLT_MODE**							**EPHY_GPIO_8_5**				**EPHY_GPIO_4_0**				
Type	RW							RW				RW				
Reset	0							0	1	1	1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						**LED_SEL**						**LED_POLARITY**				
Type						RW						RW				
Reset						0	0	0				0	0	0	0	0

Bit(s)	Name	Description
31	OLT_MODE	**EPHY OLT Mode** 0: Disable 1: Enable
24:21	EPHY_GPIO_8_5	**EPHY GPIO[8:5]** EPHY_GPIO[8:5] is used to set EPHY initial state which is latched by EPHY SW reset.
20:16	EPHY_GPIO_4_0	**EPHY GPIO[4:0]** EPHY_GPIO[4:0] is used to set EPHY MDIO address which is latched by EPHY SW reset.
10:8	LED_SEL	**LED Source** 0: ESW LED Control 1: EPHY LED Control[0] 2: EPHY LED Control[1] 3: EPHY LED Control[2]
4:0	LED_POLARITY	**Per Port LED Polarity Control** 0: Low Active 1: High Active

2.20 MSDC

2.20.1 Registers

Module name: MSDC Base address: (+10130000h)

Address	Name	Width	Register Function
10130000	**MSDC_CFG**	32	**MSDC Configuration Register** The register is for general configuration of the MS/SD controller.
10130004	**MSDC_IOCON**	32	**MSDC IO Configuration Register** The register contains the receiver path data latch timing control and interface control bits.
10130008	**MSDC_PS**	32	**MSDC Pin Status Register** The register is used to storing card detection and write protection pin status. Card detection status can be disabled.
1013000C	**MSDC_INT**	32	**MSDC Interrupt Register** The register contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled.
10130010	**MSDC_INTEN**	32	**MSDC Interrupt Enable Register** The register contains the related enable bit of interrupts.
10130014	**MSDC_FIFOCS**	32	**MSDC FIFO Control and Status Register** The register contains the control and status of embedded 128B FIFO.
10130018	**MSDC_TXDATA**	32	**MSDC TX Data Port Register** The register is for PIO mode only. Used to input MSDC write data to card. The access can be AHB 1B/2B/4B
1013001C	**MSDC_RXDATA**	32	**MSDC RX Data Port Register** The register is for PIO mode only. Used to read back MSDC read data from card. The access can be AHB 1B/2B/4B.
10130030	**SDC_CFG**	32	**SD Configuration Register** The register is used for configuring the MS/SD Memory Card Controller when it is configured as the host of SD Memory Card. If the controller is configured as the host of Memory Stick, the contents of the register have no impact on the operation of the controller.
10130034	**SDC_CMD**	32	**SD Command Register** The register defines a SD Memory Card command and its attributes. Before MS/SD controller issues a transaction onto SD bus, application shall specify other relative settings such as argument for command. After writing the register by the application, MS/SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD/MMC bus run 128 cycles before issuing the command.
10130038	**SDC_ARG**	32	**SD Argument Register** The register contains the argument of the SD/MMC Memory Card command.
1013003C	**SDC_STS**	32	**SD Status Register** The register reflects SD bus status and contains MMC stream write status.
10130040	**SDC_RESP0**	32	**SD Response Register 0** The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.
10130044	**SDC_RESP1**	32	**SD Response Register 1** The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.
10130048	**SDC_RESP2**	32	**SD Response Register 2**

			The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.
1013004C	**SDC_RESP3**	32	**SD Response Register 3** The register contains parts of the last SD/MMC Memory Card bus response. The register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 are composed of the last SD/MMC Memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of response token is stored in the register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. SDC_RESP0 = bit 31~0 SDC_RESP1 = bit 63~32 SDC_RESP2 = bit 95~64 SDC_RESP3 = bit 127~96 For response of type R1b in auto CMD12 or R1 in auto CMD23, bit 39 to 8 of response token is stored in the register field of SDC_RESP3. For the responses of other types, only bit 39 to 8 of response token is stored in the register field SDC_RESP0.
10130050	**SDC_BLK_NUM**	32	**SD Block Number Register** This register defines the block number for the block transaction. For single read/write, this register should be set to 1. For multiple read/write, this register should be set to larger than 1. Set to 0 will cause unexpected result.
10130058	**SDC_CSTS**	32	**SD Card Status Register** After commands with R1 and R1b response, this register will contain the status of the SD/MMC card
1013005C	**SDC_CSTS_EN**	32	**SD Card Status Enable Register** This register is used to control which bit of the SDC_CSTS will generate the MSDC_INT.SD_CSTA interrupt.
10130060	**SDC_DATCRC_STS**	32	**SD Card Data CRC Status Register** This register reflects the CRC status of data line[7:0]. This register is only for MSDC Read
10130080	**SD_ACMD_RESP**	32	**SD ACMD Response Register** This register stores the response of auto command from SD card
10130090	**DMA_SA**	32	**DMA Start Address Register** This register contains the start address of the DMA descriptor
10130094	**DMA_CA**	32	**DMA Current Address Register** This register contains the current DMA address
10130098	**DMA_CTRL**	32	**DMA Control Register** This register is used to control the DMA operation.
1013009C	**DMA_CFG**	32	**DMA Configuration Register** This register is used to configure the DMA operation.
101300A0	**SW_DBG_SEL**	32	**MSDC S/W Debug Selection Register** This register is used to select S/W debug output
101300A4	**SW_DBG_OUT**	32	**MSDC S/W Debug Output Register** This register shows the selected debug output
101300A8	**DMA_LENGTH**	32	**DMA Length Register** This register is used to set Basic DMA operation length
101300B0	**PATCH_BIT0**	32	**MSDC Patch Bit Register 0** This register can configure the patch function. For normal function, these bit should keep in default value
101300B4	**PATCH_BIT1**	32	**MSDC Patch Bit Register 1** This register can configure the patch function. For normal function, these bit should keep in default value
101300EC	**PAD_TUNE**	32	**MSDC Pad Tuning Register** This register can configure the delay line embedded in Pad Macro

101300F0	**DAT_RD_DLY0**	32	**MSDC Data Delay Line Register 0** This register can configure the delay line embedded in Pad Macro
101300F4	**DAT_RD_DLY1**	32	**MSDC Data Delay Line Register 1** This register can configure the delay line embedded in Pad Macro
101300F8	**HW_DBG_SEL**	32	**MSDC H/W Debug Selection Register** This register can select the H/W debug output
10130100	**MAIN_VER**	32	**MSDC Main Version Register** This register shows the version code of MSDC IP
10130104	**ECO_VER**	32	**MSDC ECO Version Register** This register shows the ECO version code of MSDC IP

10130000 MSDC_CFG MSDC Configuration Register 00000099

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																CCKMD
Type																RW
Reset																0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CCKDIV								CCKSB			CCKDRVE	PIO	RST	CCKPD	MSDC
Type	RW								RU			RW	RW	A0	RW	RW
Reset	0	0	0	0	0	0	0	0	1			1	1	0	0	1

Bit(s)	Mnemonic	Name	Description
16	**CCKMD**	CARD_CK_MODE	**MS/SD Card clock mode** 1'b0: Use clock divider output which divided by msdc_src_ck as msdc_ck, bit[15]~bit[8] should be programmed. 1'b1: Use msdc_src_ck as msdc_ck, bit[15]~bit[8] is ignored.
15:8	**CCKDIV**	CARD_CK_DIV	**MS/SD Card clock divider** The register field controls clock frequency of serial clock on MS/SD bus. Please refer to Data Line Latching Timing Diagram and Response Latching Timing Diagram. For non-DDR mode, msdc_ck equals SD bus clock. (Ex: For SDR25 or HS, msdc_ck and SD bus clock will be 50MHz) For DDR mode, msdc_ck denotes the MSDC internal clock which will be dobule to SD bus clock. (Ex: For DDR50, msdc_ck should be set to 100MHz and bus clock will be 50MHz) 8'b00000000: msdc_ck =(1/2) * msdc_src_ck 8'b00000001: msdc_ck = (1/(4*1)) * msdc_src_ck 8'b00000010: msdc_ck = (1/(4*2)) * msdc_src_ck 8'b00000011: msdc_ck = (1/(4*3))* msdc_src_ck 8'b00010000: msdc_ck = (1/(4*16))* msdc_src_ck 8'b11111111: msdc_ck = (1/(4*255)) * msdc_src_ck
7	**CCKSB**	CARD_CK_STABLE	**MS/SD Card clock stable or not** After programming the CARD_CK_MODE or CARD_CK_DIV, this bit will immediately go to "0" and return to "1" if stable. User should poll this register to make sure the safety control of MSDC. 1'b0: Clock output is not stable 1'b1: Clock output is stable
4	**CCKDRVE**	CARD_CK_DRV_EN	**SD/MS Card Bus Clock drive enable bit** Set this bit to 1 to enable MSDC bus clock driver.

Bit(s)	Mnemonic	Name	Description
			The default bus state depends on MSDC_CFG][1] CARD_CK_PWDN bit. If MSDC_CFG[1] CARD_CK_PWDN= 1, the default clock state is free running. If MSDC_CFG[1] CARD_CK_PWDN = 0, the default clock state is gated to 0. Set this bit to 0 will put the bus state into "tri-state". Default is 1. 1'b0: Put the clock pad into tri-state 1'b1: Enable MSDC to drive clock pad, the state of CLK depends on MSDC_CFG[1] CARD_CK_PWDN
3	**PIO**	PIO_MODE	**MS/SD PIO mode** PIO mode selection. Default is in PIO mode. 1'b0: DMA mode 1'b1: PIO mode
2	**RST**	RST	**Software reset** Writing 1 to this register will cause internal synchronous reset of MS/SD controller, and it will not reset register settings and DMA controller. The reset sequence is done when this bit goes to 0. S/W should wait this bit back to 0 after writing 1. 1'b0: MS/SD controller is not in reset state 1'b1: MS/SD controller is in reset state
1	**CCKPD**	CARD_CK_PWDN	**MSDC bus clock power down mode** This bit controls the card clock power down mode. 1'b0: Clock is gated to 0 if no command or data is transmitted. 1'b1: Clock is free running even if no command or data is transmitted. (The clock may still be stopped when MSDC write data is not enough or no space for next read data)
0	**MSDC**	MSDC	**MS/SD mode selection** The register bit is used to configure the controller as the host of Memory Stick or as the host of SD/MMC Memory card. The default value is to configure the controller as the host of Memory Stick. 1'b0: Configure the controller as the host of Memory Stick 1'b1: Configure the controller as the host of SD/MMC Memory card

10130004 **MSDC_IOCON** **MSDC IO Configuration Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name									RD7SPL	RD6SPL	RD5SPL	RD4SPL	RD3SPL	RD2SPL	RD1SPL	RD0SPL
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			WD3SPL	WD2SPL	WD1SPL	WD0SPL	WDSPLSEL	WDSPL			RDSPLSEL		DDLSEL	RDSPL	RSPL	SDR104CKS
Type			RW	RW	RW	RW	RW	RW			RW		RW	RW	RW	RW
Reset			0	0	0	0	0	0			0		0	0	0	0

Bit(s)	Mnemonic	Name	Description
23	**RD7SPL**	R_D7_SMPL	**Read data 7 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge

Bit(s)	Mnemonic	Name	Description
22	**RD6SPL**	R_D6_SMPL	**Read data 6 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
21	**RD5SPL**	R_D5_SMPL	**Read data 5 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
20	**RD4SPL**	R_D4_SMPL	**Read data 4 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
19	**RD3SPL**	R_D3_SMPL	**Read data 3 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
18	**RD2SPL**	R_D2_SMPL	**Read data 2 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
17	**RD1SPL**	R_D1_SMPL	**Read data 1 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
16	**RD0SPL**	R_D0_SMPL	**Read data 0 sample selection** This bit is only valid when bit 5 is ON 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
13	**WD3SPL**	W_D3_SMPL	**SDIO interrupt sample selection** This bit is only valid when bit 9 is ON 1'b0: Sample SDIO interrupt by external bus clock rising edge 1'b1: Sample SDIO interrupt by external bus clock falling edge
12	**WD2SPL**	W_D2_SMPL	**SDIO interrupt sample selection** This bit is only valid when bit 9 is ON 1'b0: Sample SDIO interrupt by external bus clock rising edge 1'b1: Sample SDIO interrupt by external bus clock falling edge
11	**WD1SPL**	W_D1_SMPL	**SDIO interrupt sample selection** This bit is only valid when bit 9 is ON 1'b0: Sample SDIO interrupt by external bus clock rising edge 1'b1: Sample SDIO interrupt by external bus clock falling edge
10	**WD0SPL**	W_D0_SMPL	**CRC Status and SDIO interrupt sample selection** This bit is only valid when bit 9 is ON 1'b0: Sample CRC Status and SDIO interrupt by external bus clock rising edge 1'b1: Sample CRC Status and SDIO interrupt by external bus clock falling edge
9	**WDSPLSEL**	W_D_SMPL_SEL	**Data line rising/falling latch fine tune selection in write transaction** 1'b0: All data line share one value indicated by MSDC_IOCON.W_D_SMPL 1'b1: Each data line has its own selection value indicated by Data line 0: MSDC_IOCON.W_D0_SMPL Data line 1: MSDC_IOCON.W_D1_SMPL Data line 2: MSDC_IOCON.W_D2_SMPL Data line 3: MSDC_IOCON.W_D3_SMPL

Bit(s)	Mnemonic	Name	Description
8	**WDSPL**	W_D_SMPL	**CRC Status and SDIO interrupt sample selection** 1'b0: Sample CRC Status and SDIO interrupt by external bus clock rising edge 1'b1: Sample CRC Status and SDIO interrupt by external bus clock falling edge
5	**RDSPLSEL**	R_D_SMPL_SEL	**Data line rising/falling latch fine tune selection in read transaction** 1'b0: All data line share one value indicated by MSDC_IOCON.R_D_SMPL 1'b1: Each data line has its own selection value indicated by Data line 0: MSDC_IOCON.R_D0_SMPL Data line 1: MSDC_IOCON.R_D1_SMPL Data line 2: MSDC_IOCON.R_D2_SMPL Data line 3: MSDC_IOCON.R_D3_SMPL Data line 4: MSDC_IOCON.R_D4_SMPL Data line 5: MSDC_IOCON.R_D5_SMPL Data line 6: MSDC_IOCON.R_D6_SMPL Data line 7: MSDC_IOCON.R_D7_SMPL
3	**DDLSEL**	D_DLYLINE_SEL	**Data line delay line fine tune selection** 1'b0: All data line share one delay selection value indicated by PAD_TUNE.PAD_DAT_RD_RXDLY 1'b1: Each data line has its own delay selection value indicated by Data line 0: DAT_RD_DLY0.DAT0_RD_DLY Data line 1: DAT_RD_DLY0.DAT1_RD_DLY Data line 2: DAT_RD_DLY0.DAT2_RD_DLY Data line 3: DAT_RD_DLY0.DAT3_RD_DLY Data line 4: DAT_RD_DLY1.DAT4_RD_DLY Data line 5: DAT_RD_DLY1.DAT5_RD_DLY Data line 6: DAT_RD_DLY1.DAT6_RD_DLY Data line 7: DAT_RD_DLY1.DAT7_RD_DLY
2	**RDSPL**	R_D_SMPL	**Read data sample selection** 1'b0: Sample read data by external bus clock rising edge 1'b1: Sample read data by external bus clock falling edge
1	**RSPL**	R_SMPL	**Command response sample selection** 1'b0: Sample response by external bus clock rising edge 1'b1: Sample response by external bus clock falling edge
0	**SDR104CKS**	SDR104_CLK_SEL	**SDR104 SCLK output clock control** This bit is only used when MSDC_CFG[17:16] CARD_CK_MODE is 2'b01. 1'b0: Bus clock output equals inverted msdc_src_ck 1'b1: Bus clock output equals msdc_src_ck

10130008 **MSDC_PS** **MSDC Pin Status Register** **01FF0000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**SDWP**							**CMD**	**DAT**							
Type	RU							RU	RU							
Reset	0							1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CDDBCE**														**CDSTS**	**CDEN**
Type	RW														RU	RW
Reset	0	0	0	0											0	0

Bit(s)	Mnemonic	Name	Description

Bit(s)	Mnemonic	Name	Description
31	**SDWP**	SD_WP	**Write Protection Switch status on SD Memory Card** The register bit shows the status of Write Protection Switch on SD Memory Card. There is no default reset value. The pin WP (Write Protection) is only useful while the controller is configured for SD Memory Card 1'b0: Write Protection Switch ON. It means that memory card is desired to be write-protected 1'b1: Write Protection Switch OFF. It means that memory card is writable
24	**CMD**	CMD	**Command line status** This bit reflects the command line value of MSDC bus.
23:16	**DAT**	DAT	**Data line status** This bit reflects the data line value of MSDC bus. (8-bits)
15:12	**CDDBCE**	CDDEBOUNCE	**Card detection de-bounce timer** The register field specifies the time interval for card detection de-bounce. Its default value is 0. It means that de-bounce interval is one 32KHz cycle. The interval will extend one cycle time of 32KHz by increasing the counter by 1
1	**CDSTS**	CDSTS	**Card detection status** 1'b0: Card detection pin status is logic low 1'b1: Card detection pin status is logic high
0	**CDEN**	CDEN	**Card detect enable** The register bit is used to control the card detection circuit 1'b0: Card detection is disable 1'b1: Card detection is enable

1013000C **MSDC_INT** **MSDC Interrupt Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													DMAPROTECT	GPDCSERR	BDCSERR	
Type													W1C	W1C	W1C	
Reset													0	0	0	
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SDDCRCERR	SDDTO	DMAXFDNE	SDXFCPL	SDCSTA	SDRCRCER	SDCTO	SDCRDY	SDIOIRQ	DMAQEPTY	SDACDRCRCER	SDACDCTO	SDACDCRDY		MSDCCDSC	MMCIRQ
Type	W1C	W1C	W1C	W1C	RU	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C		W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0

Bit(s)	Mnemonic	Name	Description
19	**DMAPROTECT**	DMA_PROTECT	**there is write operation to DMA start address, length, start bit or last buf bit**
18	**GPDCSERR**	GPD_CS_ERR	**GPD checksum error detected**
17	**BDCSERR**	BD_CS_ERR	**BD checksum error detected**
15	**SDDCRCE**	SD_DATA_CRCE	**SD Data CRC error interrupt**

Bit(s)	Mnemonic	Name	Description
	RR	RR	Indicates that MS/SD controller detects a CRC error after reading a block of data from the DAT line or SD/MMC signals a CRC error after writing a block of data to the DAT line. 1'b0: Otherwise 1'b1: MS/SD controller detected a CRC error after reading a block of data from the DAT line or SD/MMC signaled a CRC error after writing a block of data to the DAT line
14	**SDDTO**	SD_DATTO	**SD Data timeout interrupt** Indicates that SD/MMC controller detects a timeout condition while waiting for data token on the DAT line. This bit is for both data read and data write. For SD data read, timeout will occur when the read data is not presented. For SD data write, timeout will occur when the write data CRC status is not presented if PATCH_BIT[30] DETECT_WR_CRC_TIMEOUT = 1 1'b0: Otherwise 1'b1: SD/MMC controller detects a timeout condition while waiting for data token on the DAT line
13	**DMAXFDNE**	DMA_XFER_DONE	**DMA transfer done interrupt** The register bit indicates the status of data block transfer. 1'b0: Otherwise 1'b1: A data block was successfully transferred
12	**SDXFCPL**	SD_XFER_COMPLETE	**SD Data transfer complete interrupt** This bit indicates the transaction which contains data has completed. While performing tuning procedure (Execute Tuning is set to 1), SD_XFER_COMPLETE is not set to 1.
11	**SDCSTA**	SD_CSTA	**SD CSTA update interrupt** The register bit indicates any bit in the register SDC_CSTA is active, the register bit will be set to 1. S/W should clear the SDC_CSTA and this bit will be de-asserted automatically. 1'b0: No SD Memory Card interrupt 1'b1: SD Memory Card interrupt exists
10	**SDRCRCER**	SD_RESP_CRCERR	**SD Command CRC error interrupt** Indicates that SD/MMC controller detected a CRC error after reading a response from the CMD line. 1'b0: Otherwise 1'b1: SD/MMC controller detected a CRC error after reading a response from the CMD line
9	**SDCTO**	SD_CMDTO	**SD Command timeout interrupt** Indicates that SD/MMC controller detected a timeout condition while waiting for a response on the CMD line. 1'b0: Otherwise 1'b1: SD/MMC controller detected a timeout condition while waiting for a response on the CMD line
8	**SDCRDY**	SD_CMDRDY	**SD Command ready interrupt** For the command without response, the register bit will be 1 once the command completes on SD/MMC bus. For command with response without busy, the register bit will be 1 whenever the command is issued onto SD/MMC bus and its corresponding response is received without CRC error. For command with response with busy in DAT0, the register bit will be 1 whenever the command is issued onto SD/MMC bus and its corresponding response is received without CRC error and the DAT0 transited from busy to idle. 1'b0: Otherwise 1'b1: Command finish successfully without a CRC error

Bit(s)	Mnemonic	Name	Description
7	**SDIOIRQ**	SD_SDIOIRQ	**SD SDIO interrupt** This bit indicates the interrupt is sensed in the SDIO bus. 1'b0: No interrupt on SDIO bus 1'b1: Interrupt on SDIO bus
6	**DMAQEPTY**	DMA_Q_EMPTY	**DMA queue empty interrupt** This bit is used to indicate the current DMA queue is empty. Only for Descriptor mode and Enhance mode.
5	**SDACDRCRCER**	SD_AUTOCMD_RESP_CRCERR	**SD auto command CRC error interrupt** This bit is set when detecting a CRC error in the Auto command response.
4	**SDACDCTO**	SD_AUTOCMD_CMDTO	**SD auto command timeout interrupt** This bit is set if no response is returned within a specified cycles(64T in spec) from the end bit of Auto command.
3	**SDACDCRDY**	SD_AUTOCMD_CMDRDY	**SD auto command ready interrupt** This bit is set if auto command is executed without CRC error or time out.
1	**MSDCCDSC**	MSDC_CDSC	**MSDC Card detection status change interrupt** The register bit indicates if any interrupt for memory card insertion/removal exists. Whenever memory card is inserted or removed and card detection circuit is enabled, i.e., the register bit CDEN in the register MSDC_PS is set to 1, the register bit will be set to 1. It will be reset when the register is read. 1'b0: Otherwise 1'b1: Card is inserted or removed
0	**MMCIRQ**	MMC_IRQ	**MMC card interrupt** 1'b0: Otherwise 1'b1: indicates that MMC card interrupt event occurs

10130010 **MSDC_INTEN** **MSDC Interrupt Enable Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name													ENDMAPROTECT	ENGPDCSERR	ENBDCSERR	
Type													RW	RW	RW	
Reset													0	0	0	
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ENSDDCRCERR	ENSDDTO	ENDMAXFDNE	ENSDXFCPL	ENSDCSTA	ENSDRCRCER	ENSDCTO	ENSDCRDY	ENSDIOIRQ	ENDMAQEPTY	ENSDACDRCRCER	ENSDACDCTO	ENSDACDCRDY		ENMSDCCDSC	ENMMCIRQ
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0

Bit(s)	Mnemonic	Name	Description
19	**ENDMAPROTECT**	EN_DMA_PROTECT	**DMA protection interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
18	**ENGPDCS**	EN_GPD_CS_ER	**GPD checksum error interrupt enable**

Bit(s)	Mnemonic	Name	Description
	ERR	R	1'b0: Disable interrupt 1'b1: Enable interrupt
17	**ENBDCSERR**	EN_BD_CS_ERR	**BD checksum error interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
15	**ENSDDCRCERR**	EN_SD_DATA_CRCERR	**SD Data CRC error interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
14	**ENSDDTO**	EN_SD_DATTO	**SD Data timeout interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
13	**ENDMAXFDNE**	EN_SD_DMA_XFER_DONE	**DMA transfer done interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
12	**ENSDXFCPL**	EN_SD_XFER_COMPLETE	**SD Data transfer complete interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
11	**ENSDCSTA**	EN_SD_CSTA	**SD CSTA update interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
10	**ENSDRCRCER**	EN_SD_RESP_CRCERR	**SD Command CRC error interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
9	**ENSDCTO**	EN_SD_CMDTO	**SD Command timeout interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
8	**ENSDCRDY**	EN_SD_CMDRDY	**SD Command ready interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
7	**ENSDIOIRQ**	EN_SD_SDIOIRQ	**SD SDIO interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
6	**ENDMAQEPTY**	EN_DMA_Q_EMPTY	**DMA queue empty interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
5	**ENSDACDRCRCER**	EN_SD_AUTOCMD_RESP_CRCERR	**SD auto command CRC error interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
4	**ENSDACDCTO**	EN_SD_AUTOCMD_CMDTO	**SD auto command timeout interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
3	**ENSDACDCRDY**	EN_AUTOCMD_CMDRDY	**SD auto command ready interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
1	**ENMSDCCDSC**	EN_MSDC_CDSC	**MSDC Card detection status change interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt
0	**ENMMCIRQ**	EN_MMC_IRQ	**MMC card interrupt enable** 1'b0: Disable interrupt 1'b1: Enable interrupt

10130014 **MSDC_FIFOCS** **MSDC FIFO Control and Status Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FIFOCLR								TXFIFOCNT							
Type	A0								RU							
Reset	0								0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RXFIFOCNT							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31	**FIFOCLR**	FIFOCLR	**Embedded FIFO clear** Write this bit to 1 makes FIFO cleared. It will goes to 0 when FIFO is cleared. S/W needs to check this bit to make sure clearing FIFO sequence is done. This bit can be used when the data read/write sequence has error and need to clean the H/W FIFO.
23:16	**TXFIFOCNT**	TXFIFOCNT	**TX FIFO count for MSDC write** 8'd0: No data in FIFO 8'd1: 1bytes data in FIFO 8'd2: 2 bytes data in FIFO 8'd131: Maximum 131 bytes data in FIFO Others: reserved
7:0	**RXFIFOCNT**	RXFIFOCNT	**RX FIFO count for MSDC read** 8'd0: No data in FIFO 8'd1: 1bytes data in FIFO 8'd2: 2 bytes data in FIFO 8'd131: Maximum 131 bytes data in FIFO Others: reserved

10130018 **MSDC_TXDATA** **MSDC TX Data Port Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PIOTXDATA[31:16]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIOTXDATA[15:0]															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**PIOTXDATA**	PIO_TXDATA	**PIO mode TXDATA port** This register can be accessed by Byte or Half-word or Word. This port can only be accessed in PIO mode. Otherwise, the transaction will be discarded.

1013001C **MSDC_RXDATA** **MSDC RX Data Port Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	PIORXDATA[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PIORXDATA[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**PIORXDATA**	PIO_RXDATA	**PIO mode RXDATA port** This register can be accessed by Byte or Half-word or Word. This port can only be accessed in PIO mode. Otherwise, the transaction will be discarded.

10130030 SDC_CFG SD Configuration Register 00100000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTOC										INTBGP	SDIOIDE	SDIO		BUSWD	
Type	RW										RW	RW	RW		RW	
Reset	0	0	0	0	0	0	0	0			0	1	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															ENWKUPINS	ENWKUPSDIOINT
Type															RW	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
31:24	**DTOC**	DTOC	**Data Timeout Counter** The period from the end of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit of 1048576 serial clocks. 8'b00000000: Extend 1048576 more serial clock cycle 8'b00000001: Extend 1048576x2 more serial clock cycle 8'b00000010: Extend 1048576x3 more serial clock cycle 8'b11111111: Extend 1048576x 256 more serial clock cycle
21	**INTBGP**	INT_AT_BLOCK_GAP	**Interrupt at block Gap** This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. 1'b0: Disables interrupt detection at the block gap 1'b1: Enables interrupt detection at the block gap
20	**SDIOIDE**	SDIO_INT_DET_EN	**SDIO interrupt detection enable** This bit is to inform the SD controller to sense the SDIO interrupt 1'b0: SDIO interrupt detection is disabled 1'b1: SDIO interrupt detection is enabled if the SDIO bit is also on

Bit(s)	Mnemonic	Name	Description
19	**SDIO**	SDIO	**SDIO mode enable bit** This bit is to enable the support to sense the SDIO interrupt and disable the R4 response CRC check for SDIO card 1'b0: SDIO mode is disabled 1'b1: SDIO mode is enabled
17:16	**BUSWD**	BUSWIDTH	**Bus width configuration** This field is used to define the SD/MMC bus width 2'b00: 1 bit mode 2'b01: 4 bit mode 2'b10: 8 bit mode 2'b11: reserved
1	**ENWKUPINS**	WAKEUP_INS_EN	**Card status change wakeup event enable bit** 1'b0: Disable wakeup event for card status change 1'b1: Enable wakeup event for card status change
0	**ENWKUPSDIOINT**	WAKEUP_SDIOINT_EN	**SDIO card interrupt wakeup event enable bit** 1'b0: Disable wakeup event for SDIO card interrupt 1'b1: Enable wakeup event for SDIO card interrupt

10130034 SDC_CMD SD Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				ACMD	LEN											
Type				RW	RW											
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GOIRQ	STOP	RW	DTYPE			RSPTYP			BREAK	CMD					
Type	RW	RW	RW	RW			RW			RW	RW					
Reset	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
28	**ACMD**	AUTO_CMD	**Auto command enable** This field determines use of auto command functions. This function can be used in all modes including PIO/Basic DMA/Descriptor DMA/Enhanced Mode. There are two methods to stop Multiple-block read and write operation. (1) Auto CMD12 Enable Multiple-block read and write commands for memory require CMD12 to stop the operation. When ACMD-12 is used, MSDC issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the MSDC_INT register. The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands defined in the Part 3 File Security specification do not require CMD12. (2) Auto CMD23 Enable When ACMD-23 is used, MSDC issues a CMD23 automatically before issuing a command specified in the CMD field. The Host Controller Version 3.00 and later shall support this function. By writing the Command register, MSDC issues a CMD23 first and then issues a command specified by the CMD field in SDC_CMD register. If response errors of CMD23 are detected, the second command is not issued. A CMD23 error is indicated in the MSDC_INT register. 32-bit block count value for CMD23 is set to SDC_BLOCK_NUM register.

Bit(s)	Mnemonic	Name	Description
			1'b0: Disable Auto Command 1'b1: Enable Auto CMD12
27:16	**LEN**	LEN	**Length** The register field is used to define the length of one block in unit of byte in a data transaction of block mode or the data length in unit of byte in data transaction of byte mode. The maximal value of block length is 2048 bytes. 12'b000000000000: Reserved 12'b000000000001: Block length is 1 byte 12'b000000000010: Block length is 2 byte 12'b011111111111: Block length is 2047 byte 12'b100000000000: Block length is 2048 byte
15	**GOIRQ**	GO_IRQ	**GO_IRQ command** The register bit indicates if the command is GO_IRQ_STATE (CMD40) and used only for MMC protocol. If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 1'b0: The command is not GO_IRQ_STATE 1'b1: The command is GO_IRQ_STATE
14	**STOP**	STOP	**Stop command** The register bit indicates if the command is a stop transmission command. It should be set to 1 when CMD12 (SD/MMC) or CMD52 with I/O abort (SDIO) is to be issued. 1'b0: The command is not a stop transmission command 1'b1: The command is a stop transmission command
13	**RW**	RW	**Command read write selection** The register bit defines the command is a read command or write command. The register bit is valid only when the command will cause a transaction with data token. 1'b0: The command is a read command 1'b1: The command is a write command
12:11	**DTYPE**	DTYPE	**Data block selection** The register field defines data token type for the command. 2'b00: No data token for the command 2'b01: Single block transaction (only available in block mode) 2'b10: Multiple block transaction. (only available in block mode) 2'b11: Stream operation. It only shall be used in MMC protocol. (only available in block mode)
9:7	**RSPTYP**	RSPTYP	**Command response type** 3'b000: This command has no response. 3'b001: The command has R1/R5/R6/R7 response. The response token is 48-bit with CRC check (For SD/MMC/SDIO) (Not include the SDIO abort command) 3'b010: The command has R2 response. The response token is 136-bit (For SD/MMC) 3'b011: The command has R3 response. The response token is 48-bit response, no CRC check (For SD/MMC) 3'b100: The command has R4 response. The response token is 48-bit without CRC check (For SDIO) The response token is 48-bit with CRC check (For MMC) 3'b111: The command has R1b response. The response token is 48-bit (For SD/MMC/SDIO)
6	**BREAK**	BREAK	**Abort a pending MMC GO_IRQ command** It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response. 1'b0: Not a beak command 1'b1: Break a pending MMC GO_IRQ_MODE command in the controller.

Bit(s)	Mnemonic	Name	Description
5:0	**CMD**	CMD	**SD Memory Card command**

10130038 SDC_ARG SD Argument Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ARG[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ARG[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**ARG**	ARG	**Memory card controller argument register**

1013003C SDC_STS SD Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**MMCSWRCPL**															**CMD_WR_BUSY**
Type	RU															W1C
Reset	0															0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**CMDBSY**	**SDCBSY**
Type															RU	RU
Reset															0	0

Bit(s)	Mnemonic	Name	Description
31	**MMCSWRCPL**	MMC_STREAM_WR_COMPL	**MMC Stream mode write data is all flushed to MMC card** S/W can use this bit to confirm last write data are flushed to MMC then issue STOP command. This bit is only valid when the command SDC_CMD.DTYPE=2'b11. 1'b0: Last Data are partially inside MSDC 1'b1: Last data are flushed to MMC card
16		CMD_WR_BUSY	
1	**CMDBSY**	CMDBUSY	**SD Command line busy status** S/W should always read this bit to make sure the command line is not busy before sending the next command. If the command is R1B or data read/write command, S/W should check SDCBUSY bit too. Note: When Auto command 12 is enabled, this bit will be asserted immediately after SDC_CMD is written and de-asserted after auto-command 12 finishes. 1'b0: No transmission is going on CMD line on SD bus 1'b1: There exists transmission going on CMD line on SD bus
0	**SDCBSY**	SDCBUSY	**SD controller busy status**

Bit(s)	Mnemonic	Name	Description
			1'b0: SD controller is idle 1'b1: SD controller is busy

10130040 **SDC_RESP0** **SD Response Register 0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP0[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP0[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**RESP0**	RESP0	**Memory card controller response register 0**

10130044 **SDC_RESP1** **SD Response Register 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP1[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP1[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**RESP1**	RESP1	**Memory card controller response register 1**

10130048 **SDC_RESP2** **SD Response Register 2** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP2[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP2[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**RESP2**	RESP2	**Memory card controller response register 2**

1013004C **SDC_RESP3** **SD Response Register 3** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	RESP3[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP3[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**RESP3**	RESP3	**Memory card controller response register 3**

10130050 SDC_BLK_NUM SD Block Number Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BLKNUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BLKNUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
31:0	**BLKNUM**	BLOCK_NUMBER	**Memory card controller Block number** This field indicates the block number of data transaction. 32'd0: Reserved 32'd1: 1 data block 32'd2: 2 data block 32'd3: 3 data block 32'hffffffff: 4GB-1 data block

10130058 SDC_CSTS SD Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSTS[31:16]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSTS[15:0]															
Type	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**CSTS**	CSTS	**CSTA** The card status field in the response R1 or R1b field. Each bit can be write 1 clear individually.

1013005C SDC_CSTS_EN SD Card Status Enable Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSTS_EN[31:16]															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CSTS_EN[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**CSTS_EN**	CSTS_EN	**CSTA_EN** This register is used to control which bit of the CSTA will generate the MSDC_INT.SDCSTA

10130060 **SDC_DATCRC_STS** **SD Card Data CRC Status Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DCSSP**							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**DCSSP**	DAT_CRCSTS_POS	**MSDC read DATA CRC status** This register reflects the CRC status of data line[7:0]. This register is only for MSDC Read. 1'b0: No CRC error 1'b1: CRC error

10130080 **SD_ACMD_RESP** **SD ACMD Response Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ACMDRESP[31:16]**															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ACMDRESP[15:0]**															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**ACMDRESP**	AUTOCMD_RESP	**SD Auto command response register** This register stores the response[39:8] of ACMD12/ACMD23/ACMD19.

10130090 **DMA_SA** **DMA Start Address Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DMASA[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	**DMASA[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**DMASA**	DMA_STR_ADDR	**The start address of the DMA address** This register is used to set the start address of the DMA. In DMA basic mode, this field indicates the source or destination address of the data transfer which depends on the command. In descriptor base DMA, this is the descriptor chain start address.

10130094 DMA_CA DMA Current Address Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DMACA[31:16]**															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**DMACA[15:0]**															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**DMACA**	DMA_CURR_ADDR	**The current address of the DMA address** This register is used to read the current address of the DMA descriptor chain.

10130098 DMA_CTRL DMA Control Register 00006000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**BSTSZ**			**SPLIT1K**	**LASTBF**	**DMAALIGN**	**DMAMOD**						**DMARSM**	**DMASTOP**	**DMASTART**
Type		RW			RW	RW	RW	RW						WO	A0	WO
Reset		1	1	0	0	0	0	0						0	0	0

Bit(s)	Mnemonic	Name	Description
14:12	**BSTSZ**	BURST_SIZE	**DMA burst size** This field is used to specify the maximum transfer bytes allowed at the device per DMA burst. This field can not be modified when the DMA status is 1. 3'd3: 8 Bytes 3'd4: 16 Bytes 3'd5: 32 Bytes 3'd6: 64 Bytes Other: Reserved
11	**SPLIT1K**	DMA_SPLIT_1K	**This field is used to specify whether split burst when corss 1K boundry address** 1'b0: 1K boundary not split

Bit(s)	Mnemonic	Name	Description
			1'b1: 1K boundary split
10	**LASTBF**	LAST_BUF	**Last buffer of the basic DMA mode** This field indicates the last buffer in the basic DMA mode
9	**DMAALIGN**	DMA_ALIGN	**This field is used to specify whether address aligment burst size** 1'b0: do not DAM burst size alignment 1'b1: DAM burst size alignment
8	**DMAMOD**	DMA_MODE	**DMA operation mode** This field indicates operation mode of DMA 1'b0: Basic DMA mode 1'b1: Descriptor base DMA mode
2	**DMARSM**	DMA_RESUME	**DMA resume control register** This bit is used to resume the DMA transaction. Read always return 0
1	**DMASTOP**	DMA_STOP	**DMA Stop control register** This bit is used to stop the DMA transaction. When SW issue STOP command, SW must wait this bit de-assert or DMA inactive to guarantee stop done.
0	**DMASTART**	DMA_START	**DMA start control register** This bit is used to start the DMA transaction. Read always return 0

1013009C **DMA_CFG** **DMA Configuration Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																DMACHKSUM12B
Type																RW
Reset																0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name			MSDCACTIVEEN				AHBHPROT2EN								DSCPCSEN	DMASTS
Type			RW				RW								RW	RU
Reset			0	0			0	0							0	0

Bit(s)	Mnemonic	Name	Description
16	**DMACHKSUM12B**	DMA_CHK_SUM_12B	**This register indicates GPD/BD checksum cover 16byte or 12byte** 1'b0: GPD/BD checksum cover 16byte 1'b1: GPD/BD checksum only cover 12byte
13:12	**MSDCACTIVEEN**	MSDC_ACTIVE_EN	**This register will indicate how to control msdc_active** 2'b00: dynamic control msdc_active 2'b01: msdc_active = 0 2'b10: msdc_active = 1 2'b11: Reserved
9:8	**AHBHPROT2EN**	AHB_HPROT_2_EN	**This register will determine how to control hprot_2 pin of AHB bus** AHB_HPROT_2_EN = 2'b00, and Basic DMA Mode All the write transfers of a burst will access by bufferable mode except the last burst of DMA

Bit(s)	Mnemonic	Name	Description
			AHB_HPROT2_2_EN=2'b00, and Descriptor DMA Mode all the write transfers of a burst will access by bufferable mode except HW own update transfer 2'b00: dynamic control hprot_2 2'b01: hprot_2 = 0 2'b10: hprot_2 = 1
1	**DSCPCSEN**	DMA_DSCP_CS_EN	**DMA descriptor checksum enable** This bit is used to enable or disable the descriptor checksum validation function for the descriptor. This field can not be modified when the DMA status is 1.
0	**DMASTS**	DMA_STATUS	**DMA status** This bit is used to indicate the status of the DMA. 1'b0: DMA engine is inactive 1'b1: DMA engine is active

101300A0 **SW_DBG_SEL** **MSDC S/W Debug Selection Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SWDBGSEL															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**SWDBGSEL**	DBG_SEL	**MSDC debug selection** This contain is reserved!

101300A4 **SW_DBG_OUT** **MSDC S/W Debug Output Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SWDBGO[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SWDBGO[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**SWDBGO**	DBG_OUT	**MSDC debug output** 32 bit output selected by SW_DBG_SEL register

101300A8 **DMA_LENGTH** **DMA Length Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	XFSZ[31:16]															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**XFSZ[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**XFSZ**	XFER_SIZE	**DMA total transfer size** This field is used to specify the number of DMA transfer byte required for the movement of source data through DMA. This field is only valid in basic DMA mode.

101300B0 **PATCH_BIT0** **MSDC Patch Bit Register 0** **403C0006**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PTCH31**	**PTCH30**	**PTCH29**	**PTCH28**	**PTCH27**	**PTCH26**	**PTCH22**				**PTCH18**				**PTCH17**	
Type	RW	RW	RW	RW	RW	RW	RW				RW				RW	
Reset	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PTCH15**						**INTCKS**							**PTCH02**	**PTCH01**	
Type	RW						RW							RW	RW	
Reset	0						0	0	0					1	1	

Bit(s)	Mnemonic	Name	Description
31	**PTCH31**	EN_MMC_DRV_RESP	**Enable MSDC always drives bus when output wakeup response (BREAK)** 1'b0: Disable 1'b1: Enable
30	**PTCH30**	DETECT_WR_CRC_TIMEOUT	**MSDC write data CRC phase timeout detection** 1'b0: Not detect CRC phase timeout 1'b1: detect CRC phase timeout
29	**PTCH29**	SPC_ALWAYS_PUSH	**SPC Buffer push mechanism** 1'b0: Push the buffer only when read transfer is on-going 1'b1: Always push the buffer
28	**PTCH28**	SDIO_INT_DLY_SEL	**SDIO interrupt latch time selection** 1'b0: Latch the data line value in internal SDIO interrupt period 1'b1: Latch the data line value in 1 clock delay of internal SDIO interrupt period
27	**PTCH27**	SDC_CMD_CMDFAIL_SEL	**SDIO interrupt period recovery selection** 1'b0: SDIO interrupt period will re-start after a CMD12 or CMD52 command is issued 1'b1: SDIO interrupt period whenever DAT line is not busy
26	**PTCH26**	SDC_CMD_IDRT_SEL	**SD identification response time selection** The register bit indicates if the command has a response with NID (that is, 5 serial clock cycles as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 1.0) response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD). 1'b0: Otherwise. 1'b1: The command has a response with NID response time.
25:22	**PTCH22**	SDC_CFG_WDO	**SD Write Data Output Delay**

Bit(s)	Mnemonic	Name	Description
		D	The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 4'b0000: No extend. 4'b0001: Extend one more serial clock cycle. 4'b0010: Extend two more serial clock cycles. 4'b1111: Extend fifteen more serial clock cycle.
21:18	**PTCH18**	SDC_CFG_BSY DLY	**SD R1B busy detection mode** The register field is only valid for the commands with R1b response. If the command has a response of R1b type, MS/SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if operations in SD/MMC Memory Card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, then the controller will abandon the detection. 4'b0000: No extend. 4'b0001: Extend one more serial clock cycle. 4'b0010: Extend two more serial clock cycles. 4'b1111: Extend fifteen more serial clock cycle.
17	**PTCH17**	SDIO_CFG_INTC _SEL	**SDIO Interrupt model selection** 1'b0: Only when data line [1] = 0 and then trigger SDIO interurpt event 1'b1: Only when data line [3:0] = 4'b1101 and then trigger SDIO interurpt event
15	**PTCH15**	MSDC_FIFO_RD _DIS	**MSDC RXFIFO Read Disable** 1'b0: Disable FIFO read permission to RXFIFO in PIO mode 1'b1: Enable FIFO read permission to RXFIFO in PIO mode
9:7	**INTCKS**	INT_DAT_LATCH _CK_SEL	**Internal MSDC clock phase selection** Total 8 stages, each stage can delay 1 clock period of msdc_src_ck
2	**PTCH02**	DIS_REFLECT_ CMDWR_WHEN _BSY	**Enable SD command register write montior** 1'b0: Enable monitor function 1'b1: Disable monitor function
1	**PTCH01**	EN_SDC_ODD_8 BIT_SUP	**Enable SD odd number support for 8-bit data bus** 1'b0: Disable 1'b1: Enable

101300B4 **PATCH_BIT1** **MSDC Patch Bit Register 1** **FF800009**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MSHBFCKEN	MRCTLCKEN	MWCTLCKEN	MSDCKEN	MACMDCKEN	MVOLDTCKEN	MPSCCKEN	MSPCCKEN	HGDMACKEN							
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW							
Reset	1	1	1	1	1	1	1	1	1							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name			BIAS28R0	BIAS28R1	BIAS28R2				GETCRCMARGIN	GETBUSYMARGIN	CMDTA			WRTA		

Type			RW	RW	RW				RW	RW	RW			RW		
Reset			0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit(s)	Mnemonic	Name	Description
31	**MSHBFCKEN**	MSDC_CK_SHBFF_CKEN	**msdc_src_ck clock enable bit for SHBFF** 1'b0: Disable 1'b1: Enable
30	**MRCTLCKEN**	MSDC_CK_RCTL_CKEN	**msdc_src_ck clock enable bit for RCTL** 1'b0: Disable 1'b1: Enable
29	**MWCTLCKEN**	MSDC_CK_WCTL_CKEN	**msdc_src_ck clock enable bit for WCTL** 1'b0: Disable 1'b1: Enable
28	**MSDCKEN**	MSDC_CK_SD_CKEN	**msdc_src_ck clock enable bit for SD** 1'b0: Disable 1'b1: Enable
27	**MACMDCKEN**	MSDC_CK_ACMD_CKEN	**msdc_src_ck clock enable bit for ACMD** 1'b0: Disable 1'b1: Enable
26	**MVOLDTCKEN**	MSDC_CK_VOLDET_CKEN	**msdc_src_ck clock enable bit for VOLDET** 1'b0: Disable 1'b1: Enable
25	**MPSCCKEN**	MSDC_CK_PSC_CKEN	**msdc_src_ck clock enable bit for PSC** 1'b0: Disable 1'b1: Enable
24	**MSPCCKEN**	MSDC_CK_SPC_CKEN	**msdc_src_ck clock enable bit forSPC** 1'b0: Disable 1'b1: Enable
23	**HGDMACKEN**	AHB_CK_GDMA_CKEN	**hclk_ck clock enable bit for GDMA** 1'b0: Disable 1'b1: Enable
13	**BIAS28R0**	BIAS_EXTBIAS_28NM	**28NM BIAS Controler register 0**
12	**BIAS28R1**	BIAS_EN18IO_28NM	**28NM BIAS Controler register 1**
11:8	**BIAS28R2**	BIAS_TUNE_28NM	**28NM BIAS Controler register 2**
7	**GETCRCMARGIN**	GET_CRC_MARGIN	**it will add margin for get crc status when card resp crc not match spec 2cycle from endbit** 1'b0: 8 cycle reserved for get crc status from write data crc endbit 1'b1: 16 cycle reserved for get crc status from write data crc endbit
6	**GETBUSYMARGIN**	GET_BUSY_MARGIN	**it will add margin for get busy state of data0** 1'b0: 1 cycle reserved for get busy state from src status endbit 1'b1: 3cycle reserved for get busy state from src status endbit
5:3	**CMDTA**	CMD_RSP_TA_CNTR	**CMD response turn around period** The turn around cycle = CMD_RSP_TA_CNTR + 2, Only for USH104 mode, this register should be set to 0 in non-UHS104 mode
2:0	**WRTA**	WRDAT_CRCS_TA_CNTR	**Write data and CRC status turn around period** The turn around cycle = WRDAT_CRCS_TA_CNTR + 2, Only for USH104 mode, this register should be set to 0 in non-UHS104 mode

101300EC **PAD_TUNE** **MSDC Pad Tuning Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CLKTDLY					CMDRRDLY						CMDRDLY				
Type	RW					RW						RW				
Reset	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DATRRDLY								DATWRDLY				
Type				RW								RW				
Reset				0	0	0	0	0				0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:27	**CLKTDLY**	PAD_CLK_TXDLY	**CLK Pad TX Delay Control** This register is used to add delay to CLK phase. Total 32 stages
26:22	**CMDRRDLY**	PAD_CMD_RESP_RXDLY	**CMD Response Internal Delay Line Control** This register is used to fine-tune response phase latched by MSDC internal clock Total 32 stages
20:16	**CMDRDLY**	PAD_CMD_RXDLY	**CMD Pad RX Delay Line Control** This register is used to fine-tune CMD pad macro respose latch timing Total 32 stages
12:8	**DATRRDLY**	PAD_DAT_RD_RXDLY	**DAT Pad RX Delay Line Control (for MSDC read only)** This register is used to fine-tune DAT pad macro read data latch timing Total 32 stages
4:0	**DATWRDLY**	PAD_DAT_WR_RXDLY	**Write Data Status Internal Delay Line Control** This register is used to fine-tune write status phase latched by MSDC internal clock Total 32 stages

101300F0 **DAT_RD_DLY0** **MSDC Data Delay Line Register 0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				DAT0RDDLY								DAT1RDDLY				
Type				RW								RW				
Reset				0	0	0	0	0				0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DAT2RDDLY								DAT3RDDLY				
Type				RW								RW				
Reset				0	0	0	0	0				0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
28:24	**DAT0RDDLY**	DAT0_RD_DLY	**DAT0 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages
20:16	**DAT1RDDLY**	DAT1_RD_DLY	**DAT1 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages
12:8	**DAT2RDDLY**	DAT2_RD_DLY	**DAT2 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages
4:0	**DAT3RDDLY**	DAT3_RD_DLY	**DAT3 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages

101300F4 **DAT_RD_DLY1** **MSDC Data Delay Line Register 1** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name				DAT4RDDLY								DAT5RDDLY				
Type				RW								RW				
Reset				0	0	0	0	0				0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				DAT6RDDLY								DAT7RDDLY				
Type				RW								RW				
Reset				0	0	0	0	0				0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
28:24	**DAT4RDDLY**	DAT4_RD_DLY	**DAT4 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages
20:16	**DAT5RDDLY**	DAT5_RD_DLY	**DAT5 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages
12:8	**DAT6RDDLY**	DAT6_RD_DLY	**DAT6 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages
4:0	**DAT7RDDLY**	DAT7_RD_DLY	**DAT7 Pad RX Delay Line Control (for MSDC RD)** Total 32 stages

101300F8 **HW_DBG_SEL** **MSDC H/W Debug Selection Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		DBGWSEL	DBG0SEL						DBGWTSEL		DBG1SEL					
Type		RW	RW						RW		RW					
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name			DBG2SEL						DBG3SEL							
Type			RW						RW							
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
30	**DBGWSEL**	HW_DBG_WRAP_SEL	**H/W debug output selection for wrapper** 0: Select original debug pins 1: Select wrapper debug pins
29:24	**DBG0SEL**	HW_DBG0_SEL	**H/W debug output selection**
23:22	**DBGWTSEL**	HW_DBG_WRAP_TYPE_SEL	**H/W debug output selection for wrapper** 2'd0: Select dbg_in20~dbg_in3b = DRAM_DBG 2'd1: Select dbg_in20~dbg_in3b = RISC_DBG 2'd2: Select dbg_in20~dbg_in3b = AHBM_DBG 2'd3: Select dbg_in20~dbg_in3b = AHBS_DBG
21:16	**DBG1SEL**	HW_DBG1_SEL	**H/W debug output selection**
13:8	**DBG2SEL**	HW_DBG2_SEL	**H/W debug output selection**
7:0	**DBG3SEL**	HW_DBG3_SEL	**H/W debug output selection**

10130100 **MAIN_VER** **MSDC Main Version Register** **20130530**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MAINVER[31:16]															
Type	RO															
Reset	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MAINVER[15:0]															
Type	RO															
Reset	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**MAINVER**	MAIN_VER	**Main Version**

10130104 **ECO_VER** **MSDC ECO Version Register** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ECOVER[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ECOVER[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
31:0	**ECOVER**	ECO_VER	**ECO Version**

2.21 PCI Express

2.21.1 Registers

PCI_Express Changes LOG

Revision	Date	Author	Change Log
0.1	2012/7/11	James Hu	Initialization
0.2	2013/1/21	James Hu	Integrate All doc
0.3	2013/2/18	Lancelot Lin	Add Host/iNic direct access window
0.4	2013/9/23	HG Chen	Initial for MT7628, from MT7621

Module name: PCI_Express Base address: (+10140000h)

Address	Name	Width	Register Function
10140000	PCICFG	32	PCI Configuration and Status Register
10140008	PCIINT	32	PCI Interrupt after enable mask
1014000C	PCIENA	32	PCI interrupt Enable
10140020	CFGADDR	32	CONFIG TLP ADDR register
10140024	CFGDATA	32	CONFIG TLP DATA register
10140028	MEMBASE	32	Base Address for Memory Space Window
1014002C	IOBASE	32	Base Address for IO Space window
10142010	PCIE0_BAR0SETUP	32	BAR0 Setup of PCIe0 Controller
10142018	PCIE0_IMBASEBAR0	32	Internal Memory Base Address for BAR0 Space of PCIe0
10142030	PCIE0_ID	32	Vendor and Device ID of PCIe0 Controller
10142034	PCIE0_CLASS	32	Class Code and Revision ID for PCIe0 Controller
10142038	PCIE0_SUBID	32	Sub Vendor and Device ID of PCIe0 Controller(This register is valid when PCIE_RC_MODE = 0)
10142050	PCIE0_SI_STAT	32	PCIe0 System Info Status
10142060	PCIE0_DLLECR	32	PCIe0 Data Link Layer Error Counter Register
10142064	PCIE0_ECRCCR	32	PCIe0 ECRC Counter register
10142070	PCIE0_LTSSM_DELAY	32	PCIe0 LTSSM Delay
10148000	PHY_RST	32	PHY Reset (P0)
10148004	PHY_EN	32	PHY Enable (P0)

10140000 PCICFG PCI Configuration and Status Register 021007F2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					P2P_BR_DEVNUM2				P2P_BR_DEVNUM1				P2P_BR_DEVNUM0			
Type					RW				RW				RW			
Reset					0	0	1	0	0	0	0	1	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															PCIRST	

Type															RW	
Reset															1	

Bit(s)	Name	Description
27:24	P2P_BR_DEVNUM2	**Device number setting of Virtual PCI-PCI bridge #2.**
23:20	P2P_BR_DEVNUM1	**Device number setting of Virtual PCI-PCI bridge #1.**
19:16	P2P_BR_DEVNUM0	**Device number setting of Virtual PCI-PCI bridge #0.**
1	PCIRST	**PCI reset control** Available when PCIe Controller in Host mode 1: Assert the PERST_N Pin 0: De-assert the PERST_N Pin

10140008 **PCIINT** **PCI Interrupt after enable mask** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name										PCIINT2	PCIINT1	PCIINT0				
Type										RO	RO	RO				
Reset										0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

Bit(s)	Name	Description
22	PCIINT2	**PCIe2 interrupt input in RC mode** This bit indicates the PCIe interrupt from PCIe2 slot 1: PCIe2 slot have interrupt occur 0: PCIe2 slot have no interrupt occur
21	PCIINT1	**PCIe1 interrupt input in RC mode** This bit indicates the PCIe interrupt from PCIe1 slot 1: PCIe1 slot have interrupt occur 0: PCIe1 slot have no interrupt occur
20	PCIINT0	**PCIe0 interrupt input in RC mode** This bit indicates the PCIe interrupt from PCIe0 slot 1: PCIe0 slot have interrupt occur 0: PCIe0 slot have no interrupt occur

1014000C **PCIENA** **PCI interrupt Enable** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name										PCIINT2_EN	PCIINT1_EN	PCIINT0_EN				
Type										RW	RW	RW				
Reset										0	0	0				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

Bit(s)	Name	Description
22	PCIINT2_EN	**PCIINT (PCIe2) Enable Control**

Bit(s)	Name	Description
		1: Enable PCIINT (PCIe2) interrupt 0: Disable PCIINT (PCIe2) interrupt
21	PCIINT1_EN	**PCIINT (PCIe1) Enable Control** 1: Enable PCIINT (PCIe1) interrupt 0: Disable PCIINT (PCIe1) interrupt
20	PCIINT0_EN	**PCIINT (PCIe0) Enable Control** 1: Enable PCIINT (PCIe0) interrupt 0: Disable PCIINT (PCIe0) interrupt

10140020 CFGADDR CONFIG TLP ADDR register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					EXTREGNUM				BUSNUM							
Type					RW				RW							
Reset					0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DEVICENUM					FUNNUM			REGNUM							
Type	RW					RW			RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bit(s)	Name	Description
27:24	EXTREGNUM	**Extent Register Number, only avail for PCIe**
23:16	BUSNUM	**Bus Number**
15:11	DEVICENUM	**Device Number**
10:8	FUNNUM	**Function Number**
7:2	REGNUM	**Register Number**

10140024 CFGDATA CONFIG TLP DATA register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CFG_DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CFG_DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CFG_DATA	**CONFIG_DATA Register** Write to or read from this register will generates a Configuration Cycle in Host mode.

10140028 MEMBASE Base Address for Memory Space Window 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MEMBASE															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

Bit(s)	Name	Description
31:16	MEMBASE	**Base Address for Memory Space Window**

1014002C **IOBASE** **Base Address for IO Space window** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**IOBASE**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type																
Reset																

Bit(s)	Name	Description
31:16	IOBASE	**Base Address for Memory Space Window**

10142010 **PCIE0_BAR0SETUP** **BAR0 Setup of PCIe0 Controller** **01FF0001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**BAR0MSK**															
Type	RW															
Reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**BAR0ENABLE**
Type																RW
Reset																1

Bit(s)	Name	Description
31:16	BAR0MSK	**Setup for Base Address Register BAR0** When the mask bit is '1', the corresponding address bit will be masked as a hit as if no address comparison has been made. When the mask bit is '0', the corresponding address bit will be used for address comparison to determine an address hit. Each base address register can be mapped from 64KB to 2GB. The mask bit will be ignored when the corresponding enable bit is '0'. *Please set this value before the CfgWr to BAR0, else the CFGWr to BAR0 will get unknown result. 16'h7fff: 2G Space 16'h3fff: 1G Space 16'h1fff: 512M Space 16'h0fff: 256M Space 16'h07ff: 128M Space 16'h03ff: 64M Space 16'h01ff: 32M Space(Default) 16'h00ff: 16M Space 16'h007f: 8M Space 16'h003f: 4M Space

Bit(s)	Name	Description
		16'h001f: 2M Space 16'h000f: 1M Space 16'h0007: 512K Space 16'h0003: 256K Space 16'h0001: 128K Space 16'h0000: 64K Space Other: Not Support
0	BAR0ENABLE	**to determind if the BAR0 space will be enabled according to BAR1MSK** 1: Enable. 0: Disable

10142018 **PCIE0_IMBASEBAR0** **Internal Memory Base Address for BAR0 Space of PCIe0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IMBASEBAR0[16:1]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IMBASEBAR0[0:0]															
Type	RW															
Reset	0															

Bit(s)	Name	Description
31:15	IMBASEBAR0	**Internal Memory Base address for BAR0** This register is used when CHIP behaves as a PCI Express RC. The actually internal memory address being accessed by an external PCI host can be obtained from the following formula: CHIP address begin accessed = (PCI Address - BAR0) + IMBASEBAR0. When write to this register, the related bit will take effect when the corresponding bit in BAR0MSK bit is 1 and BAR0ENABLE is 1. Internal Memory Base address for BAR0 This register is used when CHIP behaves as a PCIe RC.

10142030 **PCIE0_ID** **Vendor and Device ID of PCIe0 Controller** **08010E8D**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEVID															
Type	RW															
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VENID															
Type	RW															
Reset	0	0	0	0	1	1	1	0	1	0	0	0	1	1	0	1

Bit(s)	Name	Description
31:16	DEVID	**Device ID**
15:0	VENID	**Vendor ID**

10142034 **PCIE0_CLASS** **Class Code and Revision ID for PCIe0 Controller** **0D800000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CCODE[23:8]															
Type	RW															
Reset	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CCODE[7:0]								REVID							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:8	CCODE	**Class Code**
7:0	REVID	**Revision ID**

10142038 **PCIE0_SUBID** **Sub Vendor and Device ID of PCIe0 Controller(This register is valid when PCIE_RC_MODE = 0)** **76210E8D**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SUBSYSID															
Type	RW															
Reset	0	1	1	1	0	1	1	0	0	0	1	0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SUBVENID															
Type	RW															
Reset	0	0	0	0	1	1	1	0	1	0	0	0	1	1	0	1

Bit(s)	Name	Description
31:16	SUBSYSID	**Sub Sytem ID**
15:0	SUBVENID	**Sub Vendor ID**

10142050 **PCIE0_SI_STAT** **PCIe0 System Info Status** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																LINKUP
Type																RO
Reset																0

Bit(s)	Name	Description
0	LINKUP	**PCIe Linkup Status** 1: Linkup 0: Linkdown

10142060 **PCIE0_DLLECR** **PCIe0 Data Link Layer Error Counter Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DLLP_ERR_CNT[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DLLP_ERR_CNT[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	DLLP_ERR_CNT	**Datalink Layer Error counter register record how many times datalink layer error happened**

10142064 **PCIE0_ECRCCR** **PCIe0 ECRC Counter register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ECRC_ERR_CNT[31:16]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ECRC_ERR_CNT[15:0]															
Type	RU															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	ECRC_ERR_CNT	**ECRC Error counter register record how many times ECRC error happened**

10142070 **PCIE0_LTSSM_DELAY** **PCIe0 LTSSM Delay** **00000F0C**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					L0S_IDLE_DELAY				NFTS_TIMEOUT_DELAY							
Type					RW				RW							
Reset					1	1	1	1	0	0	0	0	1	1	0	0

Bit(s)	Name	Description
11:8	L0S_IDLE_DELAY	**Entry L0S_IDLE delay for various PHY**
7:0	NFTS_TIMEOUT_DELAY	**NFTS timeout delay for various PHY**

10148000 **PHY_RST** **PHY Reset (P0)** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																

Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**P0_PHY_RST**
Type																RW
Reset																1

Bit(s)	Name	Description
0	P0_PHY_RST	**Reset for PCIE P0 PHY** 0: Assert reset 1: De-assert reset

10148004 **PHY_EN** **PHY Enable (P0)** **00000001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**P0_PHY_EN**
Type																RW
Reset																1

Bit(s)	Name	Description
0	P0_PHY_EN	**Enable of PCIE P0 PHY** 0: Disable 1: Enable

2.22 USB Host Controller

2.22.1 Registers

USB_host Changes LOG

Revision	Date	Author	Change Log
0.1	2013/11/22	Chihlung Tsou	Initialization

Module name: USB_host Base address: (+101C0000h)

Address	Name	Width	Register Function
101C0000	HCCAPBASE	32	**HCCAPBASE** Capability Register
101C0004	HCSPARAMS	32	**HCSPARAMS** Structural Parameter
101C0008	HCCPARAMS	32	**HCCPARAMS** Capability Parameter
101C0010	USBCMD	32	**USBCMD** USB Command
101C0014	USBSTS	32	**USBSTS** USB Status
101C0018	USBINTR	32	**USBINTR** USB Interrupt Enable
101C001C	FRINDEX	32	**FRINDEX** USB Frame Index
101C0020	CTRLDSSEGMENT	32	**CTRLDSSEGMENT** 4G Segment Selector
101C0024	PERIODICLISTBASE	32	**PERIODICLISTBASE** Periodic Frame List Base Address Register
101C0028	ASYNCLISTADDR	32	**ASYNCLISTADDR** Asynchronous List Address
101C0050	CONFIGFLAG	32	**CONFIGFLAG** Configured Flag Register
101C0054	PORTSC_1_to_15	32	**PORTSC_1_to_15** Port Status/Control
101C0090	INSNREG00	32	**INSNREG00** Programmable Microframe Base Value
101C0094	INSNREG01	32	**INSNREG01_31_16** Programmable Packet Buffer OUT/IN Thresholds
101C0098	INSNREG02	32	**INSNREG02_11_0** Programmable Packet Buffer Depth
101C009C	INSNREG03	32	**INSNREG03_15** Enable L1 sleep bit in ULPI function control register
101C00A0	INSNREG04	32	**INSNREG04_31_7** ULPI Configuration
101C00A4	INSNREG05	32	**Vbusy** UTMI Configuration Hardware indicator that a write to this register has occurred and the hardware is currently processing the operation defined by the data written

101C00A8	**INSNREG06**	32	**INSNREG06_31** AHB Error Captured
101C00AC	**INSNREG07**	32	**INSNREG06_31_0** AHB Master Error Address

101C0000 HCCAPBASE HCCAPBASE 01000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HCCAPBASE[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HCCAPBASE[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit(s)	Name	Description
31:0	HCCAPBASE	**Capability Register**

101C0004 HCSPARAMS HCSPARAMS 00001111

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HCSPARAMS[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HCSPARAMS[15:0]															
Type	RO															
Reset	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

Bit(s)	Name	Description
31:0	HCSPARAMS	**Structural Parameter**

101C0008 HCCPARAMS HCCPARAMS 0000A026

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HCCPARAMS[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HCCPARAMS[15:0]															
Type	RW															
Reset	1	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0

Bit(s)	Name	Description
31:0	HCCPARAMS	**Capability Parameter**

101C0010 USBCMD USBCMD 00080B00

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	USBCMD[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	USBCMD[15:0]															
Type	RW															
Reset	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	USBCMD	**USB Command**

101C0014 **USBSTS** **USBSTS** **00001000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	USBSTS[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	USBSTS[15:0]															
Type	RW															
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	USBSTS	**USB Status**

101C0018 **USBINTR** **USBINTR** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	USBINTR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	USBINTR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	USBINTR	**USB Interrupt Enable**

101C001C **FRINDEX** **FRINDEX** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FRINDEX[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FRINDEX[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
31:0	FRINDEX	**USB Frame Index**

101C0020 **CTRLDSSEGMENT** **CTRLDSSEGMENT** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CTRLDSSEGMENT[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CTRLDSSEGMENT[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CTRLDSSEGMENT	**4G Segment Selector**

101C0024 **PERIODICLISTBASE** **PERIODICLISTBASE** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERIODICLISTBASE[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PERIODICLISTBASE[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PERIODICLISTBASE	**Periodic Frame List Base Address Register**

101C0028 **ASYNCLISTADDR** **ASYNCLISTADDR** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ASYNCLISTADDR[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ASYNCLISTADDR[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	ASYNCLISTADDR	**Asynchronous List Address**

101C0050 **CONFIGFLAG** **CONFIGFLAG** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	**CONFIGFLAG[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CONFIGFLAG[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	CONFIGFLAG	**Configured Flag Register**

101C0054 **PORTSC_1_to_15** **PORTSC_1_to_15** **00002000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**PORTSC_1_to_15[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PORTSC_1_to_15[15:0]**															
Type	RW															
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	PORTSC_1_to_15	**Port Status/Control**

101C0090 **INSNREG00** **INSNREG00** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INSNREG00_31_20**												**INSNREG00_19_14[5:2]**			
Type	RO												RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INSNREG00_19_14[1:0]**		**INSNREG00_13_12**		**INSNREG00_11_1**											**INSNREG00_0**
Type	RW		RW		RW											RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:20	INSNREG00_31_20	**Programmable Microframe Base Value** Allows you to change the microframe length value (default is microframe SOF = 125 us) to reduce the simulation time. Note: Do not enable this register for the gate-level netlist.
19:14	INSNREG00_19_14	**This field is only used for debug purposes.** In heterogeneous mode, if the per port clock gets out of sync (but sill within in ppm limits) of the phy_clk , then the per port sof counter needs some correction relative to the global sof counter. The RTL corrects itself if this happens.
13:12	INSNREG00_13_12	**This value is used as the 1-microframe counter with byte interface (8-**

Bit(s)	Name	Description
		bits).
11:1	INSNREG00_11_1	**This value is used as the 1-microframe counter with word interface (16-bits).**
0	INSNREG00_0	**Writing 1'b1 enables this register.**

101C0094 **INSNREG01** **INSNREG01_31_16** **00200020**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OUT_Threshold															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IN_Threshold															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit(s)	Name	Description
31:16	OUT_Threshold	**The OUT threshold is used to start the USB transfer** as soon as the OUT threshold amount of data is fetched from system memory. It is also used to disconnect the data fetch, if the threshold amount of space is not available in the Packet Buffer.
15:0	IN_Threshold	**The IN threshold is used to start the memory transfer** as soon as the IN threshold amount of data is available in the Packet Buffer. It is also used to disconnect the data write, if the threshold amount of data is not available in the Packet Buffer.

101C0098 **INSNREG02** **INSNREG02_11_0** **00000080**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					INSNREG02_11_0											
Type					RW											
Reset					0	0	0	0	1	0	0	0	0	0	0	0

Bit(s)	Name	Description
11:0	INSNREG02_11_0	**The value specified here is the number of DWORDs (32-bit entries).**

101C009C **INSNREG03** **INSNREG03_15** **00002001**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INS	INS	INS	INSNREG03_12_1		INS	INSNREG03_8_1									INS

	NREG03_15	**NREG03_14**	**NREG03_13**		**0**		**NREG03_9**									**NREG03_0**
Type	RW	RW	RW		RW		RW				RW					RW
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit(s)	Name	Description
15	INSNREG03_15	**During L1 sleep, when interfacing with ULPI PHY,the controller keeps the SuspendM bit 1** (Powered) in the ULPI function Control register.
14	INSNREG03_14	**This bit controls the End of Resume sequence of the EHCI host controller.**
13	INSNREG03_13	**When set to 1 (default), the core ignores the linestate checking when transmitting SOF during the SE0_NAK test mode.**
12:10	INSNREG03_12_10	**This field specifies the extra delays in phy_clks to** be added to the "Transmit to Transmit turnaround delay" value maintained in the core.
9	INSNREG03_9	**In CONFIG1 mode only**
8:1	INSNREG03_8_1	**This value indicates the additional number of** bytes to be accommodated for the time-available calculation.
0	INSNREG03_0	**- 1'b1: Enables this function** - 1'b0: Disables this function

101C00A0 **INSNREG04** **INSNREG04_31_7** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RSV[24:9]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RSV[8:0]**									**INSNREG04_6**	**INSNREG04_5**	**INSNREG04_4**		**INSNREG04_2**	**INSNREG04_1**	**INSNREG04_0**
Type	RO									RW	RW	RW		RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0

Bit(s)	Name	Description
31:7	RSV	
6	INSNREG04_6	**This field is set to all** 1s by default. This field is valid if there is at least one port with ULPI interface
5	INSNREG04_5	**the automatic feature is** enabled. The Suspend signal is deasserted (logic level 1'b1) when run/stop is reset by software, but the hchalted bit is not yet set. 1'b1: Disables the automatic feature
4	INSNREG04_4	**1'b1: NAK reload fix disabled. (Incorrect NAK** reload transition at the end of a microframe for backward compatibility with Release 2.40c. For more information see the USB 2.0 Host-AHB Release Notes.

Bit(s)	Name	Description
		Reset value is 1'b0. Attribute is R/W.
2	INSNREG04_2	**Scales down port enumeration time.** Reset value is 1'b0.
1	INSNREG04_1	**The HCCPARAMS register's bits 17,** 15:4, and 2:0 become writable. Upon system reset, these bits are 0.
0	INSNREG04_0	**The HCSPARAMS register becomes** writable. Upon system reset, this bit is 0.

101C00A4 **INSNREG05** **Vbusy** **00021000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															Vbusy	VPort[3:3]
Type															RO	RO
Reset															1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VPort[2:0]			VControlLoadM	Vcontrol				Vstatus							
Type	RO			RO	RO				RO							
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
17	Vbusy	**Hardware indicator** that a write to this register has occurred and the hardware is currently processing the operation defined by the data written. When processing is finished, this bit is cleared.
16:13	VPort	**Valid values range** from 1 to 15 depending on coreConsultant configuration.
12	VControlLoadM	**- 1'b0: Load** - 1'b1: NOP, (Software R/W)
11:8	Vcontrol	
7:0	Vstatus	

101C00A8 **INSNREG06** **INSNREG06_31** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INSNREG06_31															
Type	RW															
Reset	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					INSNREG06_11_9			INSNREG06_8_4					INSNREG06_3_0			

Type					RO			RO					RO			
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	INSNREG06_31	**Indicator that an AHB error was encountered and** values were captured. To clear this field the application must write a 0 to it.
11:9	INSNREG06_11_9	**(RO) HBURST Value of the control phase** at which the AHB error occurred.
8:4	INSNREG06_8_4	**(RO) Number of beats expected in the** burst at which the AHB error occurred. Valid values are 0 to 16. - 5'b10001 - 5b11111: Reserved - 5'b00000 - 5b10000: Valid
3:0	INSNREG06_3_0	**Number of successfully-completed** beats in the current burst before the AHB error occurred.

101C00AC **INSNREG07** **INSNREG06_31_0** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**INSNREG06_31_0[31:16]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**INSNREG06_31_0[15:0]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31:0	INSNREG06_31_0	**(RO) AHB address of the control phase at** which the AHB error occurred

3. List

Abbrev.	Description
AC	Access Category
ACK	Acknowledge/ Acknowledgement
ACL	Access Control List
ACPR	Adjacent Channel Power Ratio
AD/DA	Analog to Digital/Digital to Analog converter
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
AFC	Automatic Frequency Calibration
AGC	Auto Gain Control
AIFS	Arbitration Inter-Frame Space
AIFSN	Arbitration Inter-Frame Spacing Number
ALC	Automatic Level Control
A-MPDU	Aggregate MAC Protocol Data Unit
A-MSDU	Aggregation of MAC Service Data Units
AP	Access Point
ASIC	Application-Specific Integrated Circuit
ASME	American Society of Mechanical Engineers
ASYNC	Asynchronous
BA	Block Acknowledgement
BAC	Block Acknowledgement Control
BAR	Base Address Register
BBP	Baseband Processor
BGSEL	Band Gap Select
BIST	Built-In Self-Test
BSC	Basic Spacing between Centers
BJT	Bipolar Junction Transistor
BSSID	Basic Service Set Identifier
BW	Bandwidth
CAS	Column Address Strobe
CCA	Clear Channel Assessment
CCK	Complementary Code Keying
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CCX	Cisco Compatible Extensions
CF-END	Control Frame End
CF-ACK	Control Frame Acknowledgement
CLK	Clock
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSR	Control Status Register

Abbrev.	Description
CTS	Clear to Send
CW	Contention Window
CWmax	Maximum Contention Window
CWmin	Minimum Contention Window
DAC	Digital-To-Analog Converter
DCF	Distributed Coordination Function
DDONE	DMA Done
DDR	Double Data Rate
DFT	Discrete Fourier Transform
DIFS	DCF Inter-Frame Space
DMA	Direct Memory Access
DQ	DRAM Data
DQS	Data Strobe
DSCP	Differentiated Services Code Point
DSP	Digital Signal Processor
DW	DWORD
EAP	Expert Antenna Processor
ED	Energy Detection
EDCA	Enhanced Distributed Channel Access
EECS	EEPROM chip select
EEDI	EEPROM data input
EEDO	EEPROM data output
EEPROM	Electrically Erasable Programmable Read-Only Memory
eFUSE	electrical Fuse
EESK	EEPROM source clock
EIFS	Extended Inter-Frame Space
EIV	Extend Initialization Vector
EVM	Error Vector Magnitude
FDS	Frequency Domain Spreading
FEM	Front-End Module
FEQ	Frequency Equalization
FIFO	First In First Out
FSM	Finite-State Machine
GDM	GTP Director Module
GEM	GPON Encapsulation Method
GF	Green Field
GND	Ground
GP	General Purpose
GPO	General Purpose Output
GPON	Gigabit Passive Optical Network
GPIO	General Purpose Input/Output
GPRS	General Packet Radio Service

Abbrev.	Description
GTP	GPRS Tunneling Protocol
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HT	High Throughput
HTC	High Throughput Control
I	In phase
ICV	Integrity Check Value
IFS	Inter-Frame Space
iNIC	Intelligent Network Interface Card
IV	Initialization Vector
I^2C	Inter-Integrated Circuit
I^2S	Integrated Inter-Chip Sound
I/O	Input/Output
IPI	Idle Power Indicator
IQ	In phase/Quadrature phase
JEDEC	Joint Electron Devices Engineering Council
JTAG	Joint Test Action Group
kbps	kilo (1000) bits per second
KB	Kilo (1024) Bytes
LCP	Linear Complementarity Problem
LDO	Low-Dropout Regulator
LDODIG	LDO for DIGital part output voltage
LED	Light-Emitting Diode
LTSSM	Link Training and Status State Machine
LNA	Low Noise Amplifier
LO	Local Oscillator
L-SIG	Legacy Signal Field
MAC	Medium Access Control
MCU	Microcontroller Unit
MCS	Modulation and Coding Scheme
MDC	Management Data Clock
MDIO	Management Data Input/Output
MEM	Memory
MFB	MCS Feedback
MFS	MFB Sequence
MIC	Message Integrity Code
MIMO	Multiple-Input Multiple-Output
MLD	Multicast Listener Discovery
MLNA	Monolithic Low Noise Amplifier
MM	Mixed Mode
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MPDU	MAC Protocol Data Units
MSB	Most Significant Bit

Abbrev.	Description
NAV	Network Allocation Vector
NAS	Network-Attached Server
NAT	Network Address Translation
NDP	Null Data Packet
NVM	Non-Volatile Memory
OCP	Open Core Protocol
ODT	On-die Termination
Oen	Output Enable
OFDM	Orthogonal Frequency-Division Multiplexing
OoS	Out-of-Service
OSC	Open Sound Control
PA	Power Amplifier
PAPE	Provider Authentication Policy Extension
PBC	Push Button Configuration
PBF	Packet Buffer
PCB	Printed Circuit Board
PCF	Point Coordination Function
PCM	Pulse-Code Modulation
PD	Preamble Detection
PFD	Phase-Frequency Detector
PHY	Physical Layer
PIFS	PCF Interframe Space
PLCP	Physical Layer Convergence Protocol
PLL	Phase-Locked Loop
PME	Physical Medium Entities
PMU	Power Management Unit
PN	Packet Number
PPLL	Programmable PLL
PROM	Programmable Read-Only Memory
PSDU	Physical layer Service Data Unit
PSI	Power supply Strength Indication
PSM	Power Save Mode
PTN	Packet Transport Network
QoS	Quality of Service
Q	Quadrature
R2P	Rbus to Pbus
RDG	Reverse Direction Grant
RAM	Random Access Memory
RC	Root Complex
RF	Radio Frequency
RGMII	Reduced Gigabit Media Independent Interface
RH	Relative Humidity
RoHS	Restriction on Hazardous Substances

Abbrev.	Description
ROM	Read-Only Memory
ROS	Rx Offset
RSSI	Received Signal Strength Indication (Indicator)
RTS	Request to Send
RvMII	Reverse Media Independent Interface
Rx	Receive
RXD	Received Data
RXINFO	Receive Information
RXWI	Receive Wireless Information
S	Stream
SDHC	Secure Digital High Capacity
SDIO	Secure Digital Input Output
SDRAM	Synchronous Dynamic Random Access Memory
SEC	Security
SGI	Short Guard Interval
SIFS	Short Inter-Frame Space
SoC	System-on-a-Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSCG	Spread Spectrum Clock Generator
STBC	Space–Time Block Code
SW	Switch Regulator
TA	Transmitter Address
TBTT	Target Beacon Transmission Time
TDLS	Tunnel Direct Link Setup
TKIP	Temporal Key Integrity Protocol
TOS	Tx Offset

Abbrev.	Description
TRSW	Tx/Rx Switch
TSF	Timing Synchronization Function
TSSI	Transmit Signal Strength Indication
Tx	Transmit
TxBF	Transmit Beamforming
TXD	Transmitted Data
TXDAC	Transmit Digital-Analog Converter
TXINFO	Transmit Information
TXOP	Opportunity to Transmit
TXWI	Tx Wireless Information
UART	Universal Asynchronous Rx/ Tx
USB	Universal Serial Bus
UTIF	Universal Test Interface
VGA	Variable Gain Amplifier
VCO	Voltage Controlled Oscillator
VIH	High Level Input Voltage
VIL	Low Level Input Voltage
VoIP	Voice over IP
VPID	Virtual Path Identifier
WCID	Wireless Client Identification
WEP	Wired Equivalent
WI	Wireless Information
WIV	Wireless Information Valid
WMM	Wi-Fi Multimedia
WPA	Wi-Fi Protected Access
WPDMA	Wireless Polarization Division Multiple Access
WS	Word Select

Mediatek Wi-Fi AP Software Programming Guide

Version: 4.4
Release date: 2014-10-27

Document Revision History

Revision	Date	Author	Description
1.0	2012/11/08	Pan Liu	Initial Version
1.1	2012/11/13	Pan Liu	Update iwpriv command
1.2	2012/12/11	Pan Liu	Add NoForwardingMBCast
1.3	2013/01/04	Pan Liu	Add VHT_BW and VhtBW
1.4	2013/1/14	Pan Liu	Update Apclient WPS command sample
1.5	2013/1/22	Pan Liu	Add FAQ- FixTxMode iwpriv command sample
1.6	2013/1/23	Pan Liu	Add new DAT item VHT_DisallowNonVHT and SingleSKU.dat sample.
1.7	2013/3/6	Pan Liu	Add MAC Repeater section
1.8	2013/3/8	Pan Liu	Add command and profile, DFS debug example
1.9	2013/3/13	Pan Liu	Add Singlesku.dat 5G and 2.4G sample profile and DFS example update
2.0	2013/3/15	Pan Liu	Add IgmpAdd1, WPS command and NEW BSSID Mode MAC address limitation. Update BGProtection
2.1	2013/3/27	Pan Liu	Add EfuseUploadToHost
2.2	2013/3/28	Pan Liu	Add FAQ for TX/RX unbalance issue.
2.3	2013/4/23	Pan Liu	Add iNIC system address configuration for WLAN profile support
2.4	2013/4/23	Pan Liu	Add iwpriv command AP2040Rescan, WLAN profile updates
2.5	2013/5/27	Pan Liu	Add WLAN profile and iwpriv parameters for VHT support.
2.6	2013/6/20	Pan Liu	Update WirelessMode=15, correct NoForwardingMBCast, Add AutoChannelSkipList
2.7	2013/7/4	Pan Liu	Add WLAN profile “EtherTrafficBand”
2.8	2013/7/26	Pan Liu	Add iNIC only profile and iwpriv command
2.9	2013/8/23	Pan Liu	Add iNIC only profile IsolateCard, EnhanceMultiClient, and BGMultiClient.
3.0	2013/8/29	Pan Liu	Add iwpriv command fpga_on, dataphy, databw, databasize, datagi, dataldpc for vht mode data rate setting.
3.1	2013/9/03	Pan Liu	Correct TYPO on DisConnectAllSta
3.2	2013/10/03	Pan Liu	Add VHT MCS table in Q&A
3.3	2013/11/20	Pan Liu	Update Multiple Radiu server usage
3.4	2014/01/08	Pan Liu	Add iNIC only new profile parameters
3.5	2014/01/20	Pan Liu	Update iwpriv commands and APClient command example
3.6	2014/02/11	Pan Liu	Add note for WpaMixPairCipher
3.7	2014/02/27	Pan Liu	Add iwpriv command ApCliAutoConnect and update SiteSurvey
3.8	2014/03/07	Pan Liu	Remove RadioOn from profile SoftAP is not support this option
3.9	2014/03/07	Pan Liu	Add iNIC profile TX&RTS retry counter and EDCCA profile
4.0	2014/04/01	Pan Liu	Update BADeline, datamcs and FixTxMode iwpriv command samples
4.1	2014/05/29	Hughes Kang	Add EDCCA testing
4.2	2014/07/01	Hughes Kang	Add HT_PROTECT, BASetup, BAOriTearDown, BARecTearDown, HT_TxStream, HT_RxStream, HtTxStream, HtRxStream, EntryLifeCheck, WAPI related parameters,

			WscStop
4.3	2014/09/16	Hughes Kang	Add PMF
4.4	2014/10/24	Money Wang	Update ● WDS ● WMM ● PMF ● Security ● AP-Client ● MAC Repeater ● IGMP Snooping ● MBSSID ● How to Fix Data Rate ● FAQ

Table of Contents

1 Introduction

This document is a software programming guide for Mediatek Wi-Fi SoftAP driver and it teaches you how to configure your own settings. We do provide two kinds of configuration method, profile and iwpriv. Later we show you the profile parameter list, the iwpriv command list, and some OID examples to demonstrate how to fully utilize the WLAN driver.

2 WLAN SoftAP Driver Profile

2.1 Sample Profile

```
#The word of "Default" must not be removed
Default
CountryRegion=5
CountryRegionABand=7
CountryCode=TW
BssidNum=1
SSID=RT2860AP
WirelessMode=9
TxRate=0
Channel=11
BasicRate=15
BeaconPeriod=100
DtimPeriod=1
TxPower=100
DisableOLBC=0
BGProtection=0
TxAntenna=
RxAntenna=
TxPreamble=0
RTSThreshold=2347
FragThreshold=2346
TxBurst=1
PktAggregate=0
TurboRate=0
WmmCapable=0
APSDCapable=0
DLSCapable=0
APAifsn=3;7;1;1
APCwmin=4;4;3;2
APCwmax=6;10;4;3
APTxop=0;0;94;47
APACM=0;0;0;0
BSSAifsn=3;7;2;2
BSSCwmin=4;4;3;2
BSSCwmax=10;10;4;3
BSSTxop=0;0;94;47
BSSACM=0;0;0;0
AckPolicy=0;0;0;0
NoForwarding=0
NoForwardingBTNBSSID=0
HideSSID=0
StationKeepAlive=0
ShortSlot=1
AutoChannelSelect=0
IEEE8021X=0
IEEE80211H=0
CSPeriod=10
WirelessEvent=0
IdsEnable=0
AuthFloodThreshold=32
```

```
AssocReqFloodThreshold=32
ReassocReqFloodThreshold=32
ProbeReqFloodThreshold=32
DisassocFloodThreshold=32
DeauthFloodThreshold=32
EapReqFooldThreshold=32
PreAuth=0
AuthMode=OPEN
EncrypType=NONE
RekeyInterval=0
RekeyMethod=DISABLE
PMKCachePeriod=10
WPAPSK=
DefaultKeyID=1
Key1Type=0
Key1Str=
Key2Type=0
Key2Str=
Key3Type=0
Key3Str=
Key4Type=0
Key4Str=
AccessPolicy0=0
AccessControlList0=
AccessPolicy1=0
AccessControlList1=
AccessPolicy2=0
AccessControlList2=
AccessPolicy3=0
AccessControlList3=
WdsEnable=0
WdsEncrypType=NONE
WdsList=
WdsKey=
RADIUS_Server=192.168.2.3
RADIUS_Port=1812
RADIUS_Key=ralink
own_ip_addr=192.168.5.234
EAPifname=br0
PreAuthifname=br0
HT_HTC=0
HT_RDG=0
HT_EXTCHA=0
HT_LinkAdapt=0
HT_OpMode=0
HT_MpduDensity=5
HT_BW=1
VHT_BW=1
VHT_SGI=1
VHT_STBC=0
VHT_BW_SIGNAL=0
VHT_DisallowNonVHT=0
VHT_LDPC=
HT_AutoBA=1
HT_AMSDU=0
HT_BAWinSize=64
HT_GI=1
```

HT_MCS=33
WscManufacturer=
WscModelName=
WscDeviceName=
WscModelNumber=
WscSerialNumber=

2.2 Common WLAN Profile Parameters

As you could see in *Section 2.1 Sample Profile*, all the settings obey the following syntax.

[Syntax]

Parameter=**Value**

The WLAN driver needs to be restarted after changing the profile. Otherwise, settings would not take effect and an interface down/up cycle could help.

ifconfig ra0 down
ifconfig ra0 up

2.2.1 CountryRegion

Description: Country region for WLAN radio 2.4 GHz regulation (G band)
Value:

CountryRegion=5

Region	Channels
0	1-11
1	1-13
2	10-11
3	10-13
4	14
5	1-14
6	3-9
7	5-13
31	1-14
32	1-11 active scan, 12 and 13 passive scan
33	1-14 all active scan, 14 b mode only

2.2.2 CountryRegionABand

Description: Country region for WLAN radio 5 GHz regulation (A band)
Value:

CountryRegionABand=7

Region	Channels
0	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
1	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
2	36, 40, 44, 48, 52, 56, 60, 64
3	52, 56, 60, 64, 149, 153, 157, 161
4	149, 153, 157, 161, 165

5	149, 153, 157, 161
6	36, 40, 44, 48
7	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 149, 153, 157, 161, 165
8	52, 56, 60, 64
9	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 132, 136, 140, 149, 153, 157, 161, 165
10	36, 40, 44, 48, 149, 153, 157, 161, 165
11	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 149, 153, 157, 161

2.2.3 CountryCode

Description: County code for WLAN radio regulation

Value:

CountryCode=

Note:

Default is empy.

2 characters, like TW for Taiwan.

Please refer to the following link for ISO3166 code list for other countries.

http://www.iso.org/iso/prods-services/iso3166ma/02iso-3166-code-lists/country_names_and_code_elements

This parameter can also be configured in EEPROM or eFuse.

Configuration in EEPROM or eFuse has higher priority than that in WLAN Profile.

2.2.4 ChannelGeography

Description: For Channel list builder

Value:

ChannelGeography=1

0: Outdoor
1: Indoor
2: Both

2.2.5 SSID

Description: The target BSSID string name configuration

Value:

SSID=11n-AP

0~z, 1~32 ASCII characters

2.2.6 WirelessMode

Description: Wireless mode configuration

Value:

WirelessMode=9

0: legacy 11b/g mixed

1: legacy 11B only
2: legacy 11A only
3: legacy 11a/b/g mixed
4: legacy 11G only
5: 11ABGN mixed
6: 11N only in 2.4G
7: 11GN mixed
8: 11AN mixed
9: 11BGN mixed
10: 11AGN mixed
11: 11N only in 5G
14: 11A/AN/AC mixed 5G band only (Only 11AC chipset support)
15: 11 AN/AC mixed 5G band only (Only 11AC chipset support)

2.2.7 Channel

Description: WLAN Radio channel (2.4G Band or 5G band)
Value:

Channel=0

Depends on CountryRegion or CountryRegionForABand.
Default value = 0, the driver scan BSSID's channel automatically.

2.2.8 BasicRate

Description: Basic rate support
Value:

BasicRate=15

0~4095

Note:
A bitmap represent basic support rate (A mode not support)
1: Basic rate-1Mbps
2: Basic rate-2Mbps
3: Basic rate-1Mbps, 2Mbps
4: Basic rate-5.5Mbps
15: Basic rate-1Mbps, 2Mbps, 5.5Mbps, 11Mbps
Examples:

Basic Rate Bit Map (max. 12-bit, represent max. 12 basic rates)												
Bit	11	10	9	8	7	6	5	4	3	2	1	0
Rate	54	48	36	24	18	12	9	6	11	5.5	2	1
Set	0	1	0	1	0	1	0	1	1	1	1	1
Hex	5				5				F			
Decimal	1375											

Note:
Set correct basic rates set before changing wireless mode.

11B/G Mixed, 11B/G/N Mixed, and 11N Only:
iwpriv ra0 set BasicRate=15 ➔ (0x0F: 1, 2, 5.5, 11 Mbps)

11B:
iwpriv ra0 set BasicRate=3 ➔ (0x03: 1, 2 Mbps)
11G-Only and 11G/N Mixed:
iwpriv ra0 set BasicRate=351 ➔ (0x15F: 1, 2, 5.5, 11, 6, 12, 24 Mbps)

2.2.9 BeaconPeriod

Description: Beacon period setting (It is SoftAP only)
Value:

BeaconPeriod=100

2.2.10 DtimPeriod

Description: DTIM period
Value:

DtimPeriod=1

1~255

2.2.11 TxPower

Description: WLAN Radio Transmit Power setting in percentage
Value:

TxPower=100

0~100

2.2.12 DisableOLBC

Description: Enable or disable OLBC (Overlapping Legacy BSS Condition)
Value:

DisableOLBC=0

0: disable
1: enable

2.2.13 BGProtection

Description: Enable/disable WLAN 11B or 11G protection
Value:

BGProtection=0

0: AUTO
1: On
2: Off

2.2.14 MaxStaNum

Description: Configure Maximun numbder of station that could connect with this SoftAP
Value:

MaxStaNum=0

0: disable
1~32

2.2.15 TxAntenna

Description: Configure Tx antenna number
Value:

TxAntenna=1

1: 1Tx1R
2: 2Tx2R
3: 3Tx3R

2.2.16 RxAntenna

Description: Configure Rx antenna number
Value:

RxAntenna=1

1: 1Tx1R
2: 2Tx2R
3: 3Tx3R

2.2.17 TxPreamble

Description: Enable or disable Tx preamble
Value:

TxPreamble=0

0: disable
1: enable

2.2.18 RTSThreshold

Description: Set RTS Threshold
Value:

RTSThreshold=2347

1~2347

2.2.19 FragThreshold

Description: Set Fragment threshold

Value:

FragThreshold=2346

256~2346

2.2.20 TxBurst

Description: Enable or disable Tx burst

Value:

TxBurst=1

0: disable
1: enable

2.2.21 PktAggregate

Description: Enable or disable Tx Aggregate

Value:

PktAggregate=0

0: disable
1: enable

2.2.22 NoForwarding

Description: enable or disable No forwarding STA packet within the same BSSID

Value:

NoForwarding=0

0: disable
1: enable

2.2.23 NoForwardingBTNBSSID

Description: enable or disable No Forwarding between each BSSID interface.

Value:

NoForwardingBTNBSSID=0

0: disable
1: enable

2.2.24 NoForwardingMBCast

Description: enable or disable No Forwarding multicast/broadcast packets between the same BSSID interface.

Value:

NoForwardingMBCast=0

0: disable
1: enable

2.2.25 HideSSID

Description: enable or disable Hidden SSID support

Value:

HideSSID=0

0: disable
1: enable

2.2.26 StationKeepAlive

Description: enable or disable Auto-detect the alive status of the station periodically

Value:

StationKeepAlive=0

0: disable
1~65535 seconds

2.2.27 ShortSlot

Description: enable or disable short slot time

Value:

ShortSlot=1

0: disable
1: enable

2.2.28 AutoChannelSelect

Description: Enable or disable Auto Channel Selection support

Value:

AutoChannelSelect=0

0: disable
1: Old Channel Selection Algorithm
2: New Channel Selection Algorithm

2.2.29 AutoChannelSkipList

Description: Configure channels you want to skip when Auto Channel Selection function is enabled
Value:

AutoChannelSkipList=<channel_list>

Example:

<channel_list>=2;3;4;5;7;8;10;

2.2.30 IEEE80211H

Description: enable or disable IEEE 802.11H support (DFS)
Value:

IEEE80211H=0

0: disable
1: enable

2.2.31 CSPeriod

Description: Set how many beacons with Channel Switch Announcement Element will be sent before changing a new channel.
Value:

CSPeriod=10

0 ~ 255. The default is 10.

Note: Channel switch period (Beacon count), unit is based on Beacon interval.

2.2.32 WirelessEvent

Description: enable or disable sending wireless event to the system log (Linux only)
Value:

WirelessEvent=0

0: disable
1: enable

2.2.33 IdsEnable

Description: enable or disable intrusion detection system
Value:

IdsEnable=0

0: disable
1: enable

2.2.34 AuthFloodThreshold

Description: enable or disable Authentication frame flood threshold
Value:

AuthFloodThreshold=32

0: disable
1~65535. (default=32)

2.2.35 ReassocReqFloodThreshold

Description: enable or disable Reassocation request frame flood threshold
Value:

ReassocReqFloodThreshold=32

0: disable
1~65535. (default=32)

2.2.36 ProbeReqFloodThreshold=32

Description: enable or disable Probe request frame flood threshold
Value:

ProbeReqFloodThreshold=32

0: disable
1~65535. (default=32)

2.2.37 DisassocFloodThreshold

Description: enable or disable disassocation frame flood threshold
Value:

DisassocFloodThreshold=32

0: disable
1~65535. (default=32)

2.2.38 DeauthFloodThreshold

Description: enable or disable deauthentication frame flood threshold
Value:

DeauthFloodThreshold=32

0: disable
1~65535. (default=32)

2.2.39 EapReqFooldThreshold

Description: enable or disable EAP request frame flood threshold
Value:

EapReqFooldThreshold=32

0: disable
1~65535. (default=32)

2.2.40 AccessPolicy0

Description: Set the access policy of ACL table 0.
Value:

AccessPolicy0=0

0: Disable this function
1: Allow all entries of ACL table to associate AP
2: Reject all entries of ACL table to associate AP

2.2.41 AccessControlList0

Description: Set the entry's MAC address into ACL table 0.
Value:

AccessControlList0=

[Mac Address];[Mac Address];...

Example:
00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f

Note: ACL for Bssid0, max=64

2.2.42 AccessPolicy1

Description: Set the access policy of ACL table 1.
Value:

AccessPolicy1=0

0: Disable this function
1: Allow all entries of ACL table to associate AP
2: Reject all entries of ACL table to associate AP

2.2.43 AccessControlList1

Description: Set the entry's MAC address into ACL table 1.
Value:

AccessControlList1=

[Mac Address];[Mac Address];...

Example:
00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f

Note: ACL for Bssid0, max=64

2.2.44 AccessPolicy2

Description: Set the access policy of ACL table 2.
Value:

AccessPolicy2=0

0: Disable this function
1: Allow all entries of ACL table to associate AP
2: Reject all entries of ACL table to associate AP

2.2.45 AccessControlList2

Description: Set the entry's MAC address into ACL table2.
Value:

AccessControlList2=

[Mac Address];[Mac Address];...

Example:
00:10:20:30:40:50;0A:0b:0c:0D:0e:0f;1a:2b:3c:4d:5e:6f

Note: ACL for Bssid0, max=64

2.2.46 AccessPolicy3

Description: Set the access policy of ACL table 3.
Value:

AccessPolicy3=0

0: Disable this function
1: Allow all entries of ACL table to associate AP
2: Reject all entries of ACL table to associate AP

2.2.47 AccessControlList3

Description: Set the entry's MAC address into ACL table 3.
Value:

AccessControlList3=

[Mac Address];[Mac Address];...

Example:
00:10:20:30:40:50;0a:0b:0c:0d:0e:0f;1a:2b:3c:4d:5e:6f

Note: ACL for Bssid0, max=64

2.2.48 RADIUS_Server

Description: Configure radius server IP address
Value:

RADIUS_Server=

IP address.
Example: RADIUS_Server=192.168.2.3

2.2.49 RADIUS_Port

Description: Configure radius server port number
Value:

RADIUS_Port=1812
Deafult: 1812

2.2.50 RADIUS_Key

Description: Configure radius key string
Value:

RADIUS_Key=

Example:

RADIUS_Key=ralink

2.2.51 own_ip_addr

Descripion: Configure SoftAP itself IP Address
Value:

own_ip_addr=

Example:

own_ip_addr=192.168.1.1

2.2.52 EAPifname

Description: EAPifname is assigned as the binding interface for EAP negotiation
Value:

EAPifname=

Example:

EAPifname=br0

2.2.53 PreAuthifname

Description: PreAuthifname is assigned as the binding interface for WPA2 Pre-authentication
Value:

PreAuthifname=

Example:

PreAuthifname=br0

2.2.54 HT_HTC

Description: enable or disable Support the HT control field
Value:

HT_HTC=0

0: disable
1: enable

Note: HTC Control field (4-octet) is following QoS field. An MPDU that contains the HT control field is referred to as a +HTC frame.

2.2.55 HT_RDG

Description: Enable or disable HT Reverse Direction Grant
Value:

HT_RDG=1

0: disable
1: enable

2.2.56 HT_EXTCHA

Description: To locate the 40MHz channel in combination with the control
Value:

HT_EXTCHA=0

0: Below
1: Above

2.2.57 HT_LinkAdapt

Decription: enable ir disable HT Link Adaptation Control
Value:

HT_LinkAdapt=0

0: disable
1: enable

2.2.58 HT_OpMode

Description: HT operation mode

Value:

HT_OpMode=0

0: HT mixed mode

1: HT Greenfield mode

2.2.59 HT_MpduDensity

Description: Minimum separation of MPDUs in an A-MPDU

Value:

HT_MpduDensity=4

0~7

0: no restriction
1: 1/4 µs
2: 1/2 µs
3: 1 µs
4: 2 µs
5: 4 µs
6: 8 µs
7: 16 µs

2.2.60 HT_BW

Description: HT channel bandwidth configuration

Value:

HT_BW=1

0: 20 MHz

1: 20/40 MHz

2.2.61 HT_PROTECT

Description: Enable or disable 802.11n protection mechanism

Value:

HT_PROTECT=1

0: Disable

1: Enable

2.2.62 HT_BSSCoexistence

Description: Enable or disable HT BSS coexistence support

Value:

HT_BSSCoexistence=1

0: Disable
1: Enable

2.2.63 HT_TxStream

Description: Set the number of spatial streams for transmission.
Value:

HT_TxStream=1/2/3

1~3: valid spatial streams

2.2.64 HT_RxStream

Description: Set the number of spatial streams for reception.
Value:

HT_RxStream=1/2/3

1~3: valid spatial streams

2.2.65 HT_BADecline

Description: Enable or disable decline Block Ack to peer
Value:

HT_BADecline=0

0: disable
1: enable

2.2.66 HT_AutoBA

Description: Enable or disable auto build Block Ack section with peer
Value:

HT_AutoBA=1

0: disable
1: enable

2.2.67 HT_AMSDU

Description: Enable or disable AMSDU section
Value:

HT_AMSDU=0

0: disable
1: enable

2.2.68 HT_BAWinSize

Description: Block Ack window size
Value:

HT_BAWinSize=64

1~64

2.2.69 HT_GI

Description: HT Guard interval support
Value:

HT_GI=1

0: Long guard interval
1: short guard interval

2.2.70 HT_MCS

Description: WLAN Modulation and Coding Scheme (MCS)
Value:

HT_MCS=33

0 ~15, 32: Fix MCS rate for HT rate.
33: Auto Rate Adaption, recommended

2.2.71 HT_MIMOPSMode

Description: 802.11n SM power save mode
Value:

HT_MIMOPSMode=3

0: Static SM Power Save Mode
2: Reserved
1: Dynamic SM Power Save Mode
3: SM enabled
(not fully support yet)

2.2.72 HT_DisallowTKIP

Description: Enable or disable 11N rate with 11N AP when cipher is TKIP or WEP
Value:

HT_DisallowTKIP=1

0: disable

1: enable

2.2.73 HT_STBC

Description: Enable or disable HT STBC support
Value:

HT_STBC=0

0: disable
1: enable

2.2.74 VHT_BW

Description: Enable or disable 11ac 80MHz bandwidth
Value:

VHT_BW=1

0: disable
1: enable

Note: 11AC chipset only

2.2.75 VHT_STBC

Description: Enable or disable 11ac STBC
Value:

VHT_STBC=1

0: disable
1: enable

Note: 11AC chipset only

2.2.76 VHT_BW_SIGNAL

Description: Enable or disable 11ac bandwidth signaling
Value:

VHT_BW_SIGNAL=1

0: disable
1: enable

Note: 11AC chipset only

2.2.77 VHT_LDPC

Description: Enable or disable LDPC on received packets with 11ac MCS
Value:

VHT_LDPC=1

0: disable
1: enable

Note: 11AC chipset only

2.2.78 VHT_DisallowNonVHT

Description: Enable or disable the function of rejecting connection attempt from non-VHT STA
Value:

VHT_DisallowNonVHT=1

0: disable
1: enable

Note: 11AC chipset only

2.2.79 WscManufacturer

Description: WPS manufacturer string
Value:

WscManufacturer=

Less than 64 characters

2.2.80 WscModelName

Description: WPS Mode name string
Value:

WscModelName=

Less than 32 characters

2.2.81 WscDeviceName

Description: WPS Device name string
Value:

WscDeviceName=

Less than 32 characters

2.2.82 WscModelNumber

Description: WPS Device model number string
Value:

WscModelNumber=

Less than 32 characters

2.2.83 WscSerialNumber

Description: WPS serial number string
Value:

WscSerialNumber=

Less than 32 characters

2.2.84 Wsc4digitPinCode

Description: WPS 4 digit pin code string
Value:

Wsc4digitPinCode=0

4 digit

2.2.85 VLANID

Description: set VLAN ID
Value:

VLANID=0

0: Disable

2.2.86 VLANPriority

Description: set VLAN Priority
Value:

VLANPriority=0

0: Disable

2.2.87 E2pAccessMode

Description: Select the EEPROM access mode from interface start-up
Value:

E2pAccessMode=2

0: NONE
1: EFUSE mode
2: FLASH mode
3: EEPROM mode
4: BIN FILE mode

2.2.88 EntryLifeCheck

Description: Set how many continued TX failure packets per STA can be ignored. Over the value, AP will tear down this STA, because it shall be gone.
Value:

EntryLifeCheck=20

Example:
EntryLifeCheck=1 ~ 65535. Default is 20.

2.2.89 EtherTrafficBand

Description: To bind enthernet packets with specific RF band
Value:

EtherTrafficBand=2G

2G: Bind enthernet packets with 2.4GHz RF Band
5G: Bind enthernet packets with 5GHz RF Band

Note: only available after SoftAP driver v3.0.1.2. or after version

2.3 WAPI Specific

2.3.1 Wapiifname

Description: Assign an interface name to process the WAI frame. The WAPID daemon shall be bound on this interface. If it doesn't specify, the default interface is "br0".
Value:

br0: default binding interface

2.3.2 WapiAsCertPath

Description: Assign the path of the AS certificate for the WAPI certificate authentication.
Value:

WapiAsCertPath=/etc/as.cer

2.3.3 WapiAsIpAddr

Description: Assign the IP address of the AS for the WAPI certificate authentication.
Value:

WapiAsIpAddr=192.168.222.174

2.3.4 WapiAsPort

Description: Assign the port number of the AS for the WAPI certificate authentication.
Value:

WapiAsPort=3810

2.3.5 WapiMskRekeyMethod

Description: Set the method for WAPI group key renew mechanism
Value:

DISABLE : Disable the rekey mechanism
TIME : time-based
PKT : packet-based

2.3.6 WapiMskRekeyThreshold

Description: Set the period of WAPI group key updating
Value:

0 : Disable this mechanism
10 ~ 0x3ffffff, Default is 3600.

2.3.7 WapiPsk1

Description: Set the WAPI pre-shared key
Value:

8~64 characters

2.3.8 WapiPskType

Description: Set the WAPI key type
Value:

0: HEX mode
1: ASCII mode

2.3.9 WapiUserCertPath

Description: Assign the path of the user certificate for the WAPI certificate authentication
Value:

WapiUserCertPath=/etc/user.cer

2.3.10 WapiUskRekeyMethod

Description: Set the method for WAPI unicast key renew mechanism
Value:

DISABLE : Disable the rekey mechanism
TIME : time-based
PKT : packet-based

2.3.11 WapiUskRekeyThreshold

Description: Set the period of WAPI unicast key updating
Value:

0 : Disable this mechanism
10 ~ 0x3ffffff, Default is 3600

2.4 iNIC Specific

2.4.1 Ext_LNA

Description: support External or internal LNA
Value:
Ext_LNA

0: Internal LNA
1: External LNA
Note: MT7620 iNIC driver only profile

2.4.2 Ext_PA

Description: support External or internal PA
Value:
Ext_PA

0: Internal PA
1: External PA
Note: MT7620 iNIC driver only profile

2.4.3 ExtEEPROM

Description: Support driver to read EEPROM from an external file
Value:

ExtEEPROM=1

0: read EERPM data from EEPROM chip
1: read EEPROM data from an external file

Note: The external EEPROM file must be exactly the same format as EEPROM format.
iNIC driver only profile.

2.4.4 Mem

Description: Support WLAN profile can configure iNIC system address value
Value:

Mem=addr1,value1;addr2,value2;

Example:
Mem=b0110014,ff7f5555;b011008c,2404040;
iNIC firmware will Set
1. memory address (0xb0110014) value (0xff7f5555);
2. memory address (0xb011008c) value (0x2404040);

Note: This parameter is only for iNIC driver.

2.4.5 DetectPhy

Description: Disable/Enable iNIC Phy link detection. if Phy link down will reset iNIC to load firmware.
Value:

DetectPhy=0

0: disable
1: enable

Note: only available on iNIC MT76XX FW v2.7.0.8 and after.

2.4.6 Thermal

Description: Disable/Enable iNIC thermal function
Value:

Thermal=0

0: disable
1: enable

Note:
Thermal function will be according to criteria with current temperature to configure Ant.
Criteria Value: 1~1000
default:80

Example:

iwpriv ra0 set tpc =80

Only available on iNIC MT76XX FW v2.7.0.8 and after.

2.4.7 %s_DfsSwAddCheck%d

Description: WLAN profile parameter to check DFS false alarm.
The first string is RDRegion. RDRegion string can be "CE", "FCC", "JAP", "JAP_W53", "JAP_W56".
The second integer is channel index. Channel index can be from 0 to 4.

Value:
There are four parameter (Period low, Period High, Width low, Width high) in one rule. Multiple rules can be used. At least one rule must be used. Each parameter is separated by semicolon. T_Low;T_High;W_Low;W_High

For example:

CE_DfsSwAddCheck0=100;200;50;500
FCC_DfsSwAddCheck0=100;200;50;500;70;700;30;300

Note: only available on iNIC MT76XX FW v2.7.0.8 and after.

2.4.8 IsolateCard

Description: Disable/Enable for iNIC isolate concurrent card traffic.
Value:

IsolateCard=0

0: disable
1: enable (iNIC concurrent card traffic can't forward to each other)

Note: only available on iNIC MT76XX FW v2.7.0.9 and after.

2.4.9 EnhanceMultiClient

Description: Disable/Enable multiple N client related configuration.
Value:

EnhanceMultiClient=0

0: disable
1: enable

Note: only available on iNIC MT76XX FW v2.7.0.9 and after.

2.4.10 BGMultiClient

Description: Disable/Enable multiple legacy client related configuration.
Value:

BGMultiClient=0

0: disable
1: enable

Note: only available on iNIC MT76XX FW v2.7.0.9 and after.

2.4.11 RssiDisauth

Description: Disable or Enable RSSI disassociate feature..
Value:

RssiDisauth=0

0: disable
1: enable

Default : 0 (disable);

If Enable RSSI disassociate feature.
Two scenarios for this feature:
(1.) STA was exceeded the RssiThreshold value. AP will disassociate STA.
(2.) Periodically Checking:
After client was associated. AP will check RSSI periodically base on PollingRssiInterval. If STA was exceeding the RssiThreshold. A counter will be increase. STA will be disassociated when STA's own counter was exceeded TimeExceedRssiThreshold. The counter will be reset if AP found STA didn't exceed the RSSI threshold.

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.12 RssiThreshold

Description: Minimum RSSI disassociate threshold.
Value:

RssiThreshold=0

Default : 0 (disable);
value : -100 ~ -1

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.13 PollingRssiInterval

Description: Polling time interval for check STA RSSI(in second).
Value:

PollingRssiInterval=0

Default : 0 (disable);
value : 1 ~ 3600

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.14 TimeExceedRssiThreshold

Description: Time of user exceed the RSSI threshold before disassociate
Value:

TimeExceedRssiThreshold=0

Default : 0 (disable);
value : 1 ~ 10000

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.15 SiteSurveyRssi

Description: Disable or Enable get RSSI for each Site Survey APT
Value:

SiteSurveyRssi=0

Default : 0
value : 0/1

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.16 AssociationInfoEvent

Description: Disable or Enable association send event include wireless mode/PHY rate/RSSI
Value:

AssociationInfoEvent=0

Default : 0
value : 0/1

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.17 EDCCA

Description: Disable or Enable EDCCA function
Value:

EDCCA=0

Default : 0
value : 0/1

Note: only available on iNIC MT76XX FW v2.7.1.0 and after.

2.4.18 TX_RETRY_NUM

Description: Tx retry number
Value:

TX_RETRY_NUM=3

Default:0
Note: only available on iNIC MT76XX FW v3.0.0.2 and after.

2.4.19 RTS_RETRY_NUM

Description: RTC retry number
Value:

RTS_RETRY_NUM=3

Default=0
Note: only available on iNIC MT76XX FW v3.0.0.2 and after.

2.4.20 EDCCA_AP_STA_TH

Description: STA count on SoftAP
Value:

EDCCA_AP_STA_TH=1

Default:1
Note: only available on iNIC MT76XX FW v3.0.0.2 and after.

2.4.21 EDCCA_AP_AP_TH

Description: SoftAP count on the same working channel
Value:

EDCCA_AP_AP_TH=1

Default:1
Note: only available on iNIC MT76XX FW v3.0.0.2 and after.

2.4.22 EDCCA_AP_RSSI_TH

Description: SoftAP count threshold on the same working channel, only when SofAP RSSI is greater than the configured level.
Value:

```
EDCCA_AP_RSSI_TH=-80
```

Note: only available on iNIC MT76XX FW v3.0.0.2 and after.

3 WLAN SoftAP Driver iwpriv set command

Syntax is iwpriv ra0 set [parameters]=[Value]

Note: Execute one iwpriv/set command at a time.

3.1.1 Debug

Description: config WLAN driver Debug level.
Value:

iwpriv ra0 set Debug=3

0~5
0: Debug Off
1: Debug Error
2: Debug Warning
3: Debug Trace
4: Debug Info
5: Debug Loud

3.1.2 DriverVersion

Description: Check driver version by iwpriv command. (Need to enable debug mode)
Value:

iwpriv ra0 set DriverVersion=0

Any value

3.1.3 CountryRegion

Description: Country region for WLAN radio 2.4 GHz regulation (G band)
Value:

iwpriv ra0 set CountryRegion=5

Region	Channels
0	1-11
1	1-13
2	10-11
3	10-13
4	14
5	1-14
6	3-9
7	5-13
31	1-14
32	1-11 active scan, 12 and 13 passive scan

33	1-14 all active scan, 14 b mode only

3.1.4 CountryRegionABand

Description: Country region for WLAN radio 5 GHz regulation (A band)
Value:

iwpriv rai0 set CountryRegionABand=7

Region	Channels
0	36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, 161, 165
1	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140
2	36, 40, 44, 48, 52, 56, 60, 64
3	52, 56, 60, 64, 149, 153, 157, 161
4	149, 153, 157, 161, 165
5	149, 153, 157, 161
6	36, 40, 44, 48
7	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 149, 153, 157, 161, 165
8	52, 56, 60, 64
9	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 132, 136, 140, 149, 153, 157, 161, 165
10	36, 40, 44, 48, 149, 153, 157, 161, 165
11	36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 149, 153, 157, 161

3.1.5 CountryCode

Description: County code for WLAN radio regulation
Value:

iwpriv ra0 set CountryCode=TW

Note:
2 characters, like TW for Taiwan.
Please refer to the following link for ISO3166 code list for other countries.
http://www.iso.org/iso/prods-services/iso3166ma/02iso-3166-code-lists/country_names_and_code_elements

3.1.6 AccessPolicy

Description: Configure access policy of ACL table
Value:

iwpriv ra0 set AccessPolicy=0

0: Disable this function
1: Allow all entries of ACL table to associate AP
2: Reject all entries of ACL table to associate AP

3.1.7 ResetCounter

Description:Reset all statistic counter
Value:

iwpriv ra0 set ResetCounter=1

3.1.8 SiteSurvey

Description: Make a site survey request to the driver

Value:

iwpriv ra0 set SiteSurvey=

Note:

Passive scan: Use empty string as argument, like "iwpriv ra0 set SiteSurvey="

Active scan: Use legal SSID as argument, like "iwpriv ra0 set SiteSurvey=Target_SSID"

3.1.9 CountryString

Description: configure country string

Value:

iwpriv ra0 set CountryString=TAIWAN

32 characters, ex:Taiwan, case insensitive

Note: Please refer to ISO3166 code list for other countries and can be found at http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html#sz

Item	Country Number	ISO Name	Country Name (CountryString)	Support 802.11A	802.11A Country Region	Support 802.11G	802.11G Country Region
	0	DB	Debug	Yes	A_BAND_REGION_7	Yes	G_BAND_REGION_5
	8	AL	ALBANIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	12	DZ	ALGERIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	32	AR	ARGENTINA	Yes	A_BAND_REGION_3	Yes	G_BAND_REGION_1
	51	AM	ARMENIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	36	AU	AUSTRALIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	40	AT	AUSTRIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	31	AZ	AZERBAIJAN	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	48	BH	BAHRAIN	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	112	BY	BELARUS	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	56	BE	BELGIUM	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	84	BZ	BELIZE	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	68	BO	BOLIVIA	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	76	BR	BRAZIL	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	96	BN	BRUNEI DARUSSALAM	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	100	BG	BULGARIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	124	CA	CANADA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	152	CL	CHILE	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	156	CN	CHINA	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	170	CO	COLOMBIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	188	CR	COSTA RICA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	191	HR	CROATIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	196	CY	CYPRUS	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	203	CZ	CZECH REPUBLIC	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	208	DK	DENMARK	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	214	DO	DOMINICAN REPUBLIC	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	218	EC	ECUADOR	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	818	EG	EGYPT	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	222	SV	EL SALVADOR	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	233	EE	ESTONIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	246	FI	FINLAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	250	FR	FRANCE	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	268	GE	GEORGIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	276	DE	GERMANY	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1

	300	GR	GREECE	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	320	GT	GUATEMALA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	340	HN	HONDURAS	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	344	HK	HONG KONG	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	348	HU	HUNGARY	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	352	IS	ICELAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	356	IN	INDIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	360	ID	INDONESIA	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	364	IR	IRAN	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	372	IE	IRELAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	376	IL	ISRAEL	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	380	IT	ITALY	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	392	JP	JAPAN	Yes	A_BAND_REGION_9	Yes	G_BAND_REGION_1
	400	JO	JORDAN	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	398	KZ	KAZAKHSTAN	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	408	KP	KOREA DEMOCRATIC	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
	410	KR	KOREA REPUBLIC OF	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
	414	KW	KUWAIT	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	428	LV	LATVIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	422	LB	LEBANON	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	438	LI	LIECHTENSTEIN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	440	LT	LITHUANIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	442	LU	LUXEMBOURG	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	446	MO	MACAU	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	807	MK	MACEDONIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	458	MY	MALAYSIA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	484	MX	MEXICO	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	492	MC	MONACO	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	504	MA	MOROCCO	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	528	NL	NETHERLANDS	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	554	NZ	NEW ZEALAND	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	578	NO	NORWAY	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	512	OM	OMAN	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	586	PK	PAKISTAN	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	591	PA	PANAMA	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	604	PE	PERU	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	608	PH	PHILIPPINES	Yes	A_BAND_REGION_4	Yes	G_BAND_REGION_1
	616	PL	POLAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	620	PT	PORTUGAL	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	630	PR	PUERTO RICO	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	634	QA	QATAR	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	642	RO	ROMANIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	643	RU	RUSSIA FEDERATION	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	682	SA	SAUDI ARABIA	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	702	SG	SINGAPORE	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	703	SK	SLOVAKIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	705	SI	SLOVENIA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	710	ZA	SOUTH AFRICA	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	724	ES	SPAIN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	752	SE	SWEDEN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	756	CH	SWITZERLAND	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	760	SY	SYRIAN ARAB REPUBLIC	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	158	TW	TAIWAN	Yes	A_BAND_REGION_3	Yes	G_BAND_REGION_0
	764	TH	THAILAND	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	780	TT	TRINIDAD AND TOBAGO	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	788	TN	TUNISIA	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	792	TR	TURKEY	Yes	A_BAND_REGION_2	Yes	G_BAND_REGION_1
	804	UA	UKRAINE	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	784	AE	UNITED ARAB EMIRATES	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1

	826	GB	UNITED KINGDOM	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_1
	840	US	UNITED STATES	Yes	A_BAND_REGION_0	Yes	G_BAND_REGION_0
	858	UY	URUGUAY	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
	860	UZ	UZBEKISTAN	Yes	A_BAND_REGION_1	Yes	G_BAND_REGION_0
	862	VE	VENEZUELA	Yes	A_BAND_REGION_5	Yes	G_BAND_REGION_1
	704	VN	VIET NAM	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	887	YE	YEMEN	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1
	716	ZW	ZIMBABWE	No	A_BAND_REGION_0	Yes	G_BAND_REGION_1

3.1.10 SSID

Description: Set AP SSID
Value:

iwpriv ra0 set SSID=11n-AP

0~z, 1~32 ASCII characters

3.1.11 WirelessMode

Description: Set WLAN mode
Value:

iwpriv ra0 set WirelessMode=5

0: legacy 11b/g mixed
1: legacy 11B only
2: legacy 11A only
3: legacy 11a/b/g mixed
4: legacy 11G only
5: 11ABGN mixed
6: 11N only
7: 11GN mixed
8: 11AN mixed
9: 11BGN mixed
10: 11AGN mixed
11: 11N only in 5G band only
14: 11A/AN/AC mixed 5G band only (Only 11AC chipset support)
15: 11 AN/AC mixed 5G band only (Only 11AC chipset support)

3.1.12 FixedTxMode

Description: Fix Tx mode to CCK or OFDM for MCS rate selection
Value:

iwpriv ra0 set FixedTxMode=CCK

CCK
OFDM

3.1.13 BasicRate

Description: configure basic rate

Value:

iwpriv ra0 set BasicRate=

0~4095

Basic Rate Bit Map (max. 12-bit, represent max. 12 basic rates)												
Bit	11	10	9	8	7	6	5	4	3	2	1	0
Rate	54	48	36	24	18	12	9	6	11	5.5	2	1
Set	0	1	0	1	0	1	0	1	1	1	1	1
Hex	5				5				F			
Decimal	1375											

Note: Be careful to set this value, if you don't know what this is, please don't set this field.

3.1.14 Channel

Description: Configure wireless channel

Value:

iwpriv ra0 set Channel=6

802.11b/g: 1 ~ 14 (it must agree with the CountryRegion setting)

802.11a: 36~165 (it must agree with the CountryRegionABand setting)

3.1.15 BeaconPeriod

Description: configure Beacon period

Value:

iwpriv ra0 set BeaconPeriod=100

20 ~ 1024 (unit is in milli-seconds)

3.1.16 DtimPeriod

Description: Configure DTIM period

Value:

iwpriv ra0 set DtimPeriod=1

1~5

3.1.17 TxPower

Description: Set Transmit Power by percentage

Value:

iwpriv ra0 set TxPower=100

0~100

Note:

91 ~ 100% & AUTO, treat as 100% in terms of mW

61 ~ 90%, treat as 75% in terms of mW	-1dBm
31 ~ 60%, treat as 50% in terms of mW	-3dBm
16 ~ 30%, treat as 25% in terms of mW	-6dBm
10 ~ 15%, treat as 12.5% in terms of mW	-9dBm
0 ~ 9 %, treat as MIN(~3%) in terms of mW	-12dBm

3.1.18 BGProtection

Description: Enable or disable 11B, 11G protection
Value:

iwpriv ra0 set BGProtection=0

0: disable
1: Always on
2:Always off

3.1.19 DisableOLBC

Description: enable or disable OLBC
Value:

iwpriv ra0 set DisableOLBC=0

0: disable
1: enable

3.1.20 TxPreamble

Description: enable or disable Tx preamble
Value:

iwpriv ra0 set TxPreamble=1

0: disable
1: enable

3.1.21 RTSThreshold

Description: Set RTS Threshold
Value:

iwpriv ra0 set RTSThreshold=2347

1~2347

3.1.22 FragThreshold

Description: Set Fragment threshold
Value:

iwpriv ra0 set FragThreshold=2346

256~2346

3.1.23 TxBurst

Description: enable or disable Tx burst mode
Value:

iwpriv ra0 set TxBurst=0

0: disable
1: enable

3.1.24 PktAggregate

Description: enable or disable packet aggregation (Ralink to Ralink only)
Value:

iwpriv ra0 set PktAggregate=1

0: disable
1: enable

3.1.25 NoForwarding

Description: enable or disable no forwarding packet between STAs in the same BSSID
Value:

iwpriv ra0 set NoForwarding=0

0: disable
1: enable

3.1.26 NoForwardingBTNBSSID

Description: enable or disable No Forwarding between each BSSID interface.
Value:

iwpriv ra0 set NoForwardingBTNBSSID=1

0: disable
1: enable

3.1.27 NoForwardingMBCast

Description: enable or disable No Forwarding multicast/broadcast packets between each BSSID interface.
Value:

iwpriv ra0 set NoForwardingMBCast=1

0: disable

1: enable

3.1.28 HideSSID

Description: enable or disable hidden SSID
Value:

iwpriv ra0 set HideSSID=1

0: disable
1: enable

3.1.29 ShortSlot

Description: enable or disabllle short slot time
Value:

iwpriv ra0 set ShortSlot=0

0: disable
1: enable

3.1.30 DisConnectSta

Description: Disconnect one specific STA which connected with this SoftAP manually
Value:

iwpriv ra0 set DisConnectSta=00:11:22:33:44:55

[MAC address]

3.1.31 DisConnectAllSta

Description: Disconnect all STAs which connected with this SoftAP manually.
Value:

iwpriv ra0 set DisConnectAllSta=1

1: disconnect all STAs

3.1.32 McastPhyMode

Description: Configure multicast physical mode
Value:

iwpriv ra0 set McastPhyMode=0

0: Disable
1: CCK
2: OFDM
3 HTMIX

3.1.33 McastMcs

Description: Specify the MCS of multicast packets.
Value:

iwpriv ra0 set McastMcs=0

0~15

3.1.34 WscVendorPinCode

Description: Set vendor pin code as pin code of WPS AP's enrollee
Value:

iwpriv ra0 WscVendorPinCode=xxxxxxx

xxxxxxxx //Valid PIN code

3.1.35 ACLAddEntry

Description: To insert one or several MAC addresses into Access control MAC table list, up to 64 MAC address at one time.
Value:

iwpriv ra0 set ACLAddEntry="xx:xx:xx:xx:xx:xx"

[MAC address];[MAC address];...;[MAC address]"
Example:

iwpriv ra0 set ACLAddEntry="00:0c:43:28:aa:12;00:0c:43:28:aa:11;00:0c:43:28:aa:10"

3.1.36 ACLClearAll

Description: To clear all the MAC address entries in an Access control MAC table list.
Value:

iwpriv ra0 set ACLClearAll=1

1: indicate to clear the table
Other value is invalid.

3.1.37 MaxStaNum

Description: To limit the maximum number of associated clients per BSS.
Value:

iwpriv ra0 set MaxStaNum=0

0: disable this function
1~32 (default:32)

3.1.38 AutoFallBack

Description: enable or disable auto fall back rate control function
Value:

iwpriv ra0 set AutoFallBack=1

0: disable
1: enable

3.1.39 GreenAP

Description: enable or disable Green AP fucntion
Value:

iwpriv ra0 set GreenAP=0

0: disable
1: enable

3.1.40 AutoChannelSel

Description: auto channel select when driver is loaded
Value:

iwpriv ra0 set AutoChannelSel=2

0: Disable
1: Old Channel Selection Algorithm
2: New Channel Selection Algorithm

3.1.41 ACSCheckTime

Description: Set a periodic check time for auto channel selection (unit: hour)
Value:

iwpriv ra0 set ACSCheckTime=3

0: Disable

3.1.42 MBSSWirelessMode

Description: Set MBSS Wireless phy Mode. Only support in v2.5.0.0 and after version.
Value:

0: 802.11 B/G mixed
1: 802.11 B only
2: 802.11 A only
4: 802.11 G only
6: 802.11 N only
7: 802.11 G/N mixed
8: 802.11 A/N mixed
9: 802.11 B/G/N mixed
10: 802.11 A/G/N mixed
11: 802.11 N in 5G band only

Example:
ra0: B/G/N fixed
ra1: B only
ra2: B/G mixed
ra3: G only
Must set main BSS (ra0) first then set other MBSS WirelessMode. Can't have A & B mode fixed in MBSS.

iwpriv ra0 set WirelessMode=9
iwpriv ra1 set MBSSWirelessMode=1
iwpriv ra2 set MBSSWirelessMode=0
iwpriv ra3 set MBSSWirelessMode=4

3.1.43 HwAntDiv

Description: enable or disable Hardware antenna diversity
Value:
iwpriv ra0 set HwAntDiv=0

0: disable
1: enable

Note: RT5350 only

3.1.44 HtBw

Description: HT channel bandwidth configuration
Value:
iwpriv ra0 set HtBw=1

0: 20 MHz
1: 20/40 MHz

3.1.45 VhtBw

Description: Enable or disable 11AC 80MHz Bandwidth support
Value:
iwpriv ra0 set VhtBw=1

0: disable
1: enable
Note: 11AC chipset only

3.1.46 VhtStbc

Description: Enable/disable 11AC STBC Support
Value:

iwpriv ra0 set VhtStbc=1

0: disable
1: enable
Note: 11AC chipset only

3.1.47 VhtBwSignal

Description: Enable/disable 11 AC BandWidth signaling
Value:

iwpriv ra0 set VhtBwSignal=1

0: disable
1: enable
Note: 11AC chipset only.

3.1.48 VhtDisallowNonVHT

Description: Enable/disable to reject non-VHT STA to connect
Value:

iwpriv ra0 set VhtDisallowNonVHT=1

0: disable
1: enable to reject non-VHT STA
Note: 11AC chipset only.

3.1.49 HtMcs

Description: Set WLAN Modulation and Coding Scheme (MCS)
Value:

iwpriv ra0 set HtMcs=33

0 ~15, 32: Fix MCS rate for HT rate.
33: Auto Rate Adaption, recommended

HT Mixed Mode, Refer to IEEE P802.11n Figure n67 HT Greenfield, Refer to IEEE P802.11n Figure n68	
MCS = 0 (1S)	(BW=0, SGI=0) 6.5Mbps
MCS = 1	(BW=0, SGI=0) 13Mbps
MCS = 2	(BW=0, SGI=0) 19.5Mbps
MCS = 3	(BW=0, SGI=0) 26Mbps
MCS = 4	(BW=0, SGI=0) 39Mbps
MCS = 5	(BW=0, SGI=0) 52Mbps
MCS = 6	(BW=0, SGI=0) 58.5Mbps
MCS = 7	(BW=0, SGI=0) 65Mbps
MCS = 8 (2S)	(BW=0, SGI=0) 13Mbps
MCS = 9	(BW=0, SGI=0) 26Mbps
MCS = 10	(BW=0, SGI=0) 39Mbps
MCS = 11	(BW=0, SGI=0) 52Mbps

MCS = 12	(BW=0, SGI=0) 78Mbps
MCS = 13	(BW=0, SGI=0) 104Mbps
MCS = 14	(BW=0, SGI=0) 117Mbps
MCS = 15	(BW=0, SGI=0) 130Mbps
MCS = 32	(BW=1, SGI=0) HT duplicate 6Mbps
Notes: When BW=1, PHY_RATE = PHY_RATE * 2 When SGI=1, PHY_RATE = PHY_RATE * 10/9 The effects of BW and SGI are accumulative. When MCS=0~7(1S, One Tx Stream), SGI option is supported. BW option is supported. When MCS=8~15(2S, Two Tx Stream), SGI option is supported. BW option is supported. When MCS=32, only SGI option is supported. BW option is not supported. (BW =1) Other MCS code in HT mode are reserved.	

3.1.50 HtGi

Description: Set WLAN Guard interval support
Value:

iwpriv ra0 set HtGi=1

0: long guard interval
1: short guard interval

3.1.51 HtOpMode

Description: HT operation Mode
Value:

iwpriv ra0 set HtOpMode=0

0: HT mixed mode
1: HT Greenfield mode

3.1.52 HtStbc

Description: Enable or disable HT STBC
Value:

iwpriv ra0 set HtStbc=1

0: disable
1: enable

3.1.53 HtExtcha

Description: To locate the 40MHz channel in combination with the control
Value:

iwpriv ra0 set HtExtcha=0

0: below
1: Above

3.1.54 HtMpduDensity

Description: Minimum separation of MPDUs in an A-MPDU
Value:

iwpriv ra0 set HtMpduDensity=4

0~7
0: no restriction
1: 1/4 μs
2: 1/2 μs
3: 1 μs
4: 2 μs
5: 4 μs
6: 8 μs
7: 16 μs

3.1.55 HtBaWinSize

Description: Block Ack window size
Value:

iwpriv ra0 set HtBaWinSize=64

1~64

3.1.56 HtTxBASize

Description: Set the number of AMPDU aggregation size of one transmission burst.
Value:

iwpriv ra0 set HtTxBASize=64

1~64: valid value

3.1.57 HtRdg

Description: Enable or disable HT Reverse Direction Grant
Value:

iwpriv ra0 set HtRdg=1

0: disable
1: enable

3.1.58 HtAmsdu

Description: Enable or disable AMSDU section
Value:

Iwpriv ra0 set HtAmsdu=0

0: disable
1: enable

3.1.59 HtAutoBa

Description: Enable or disable auto build Block Ack section with peer
Value:

iwpriv ra0 set HtAutoBa=1

0: disable
1: enable

3.1.60 BADecline

Description: Enable or disable decline Block Ack to peer
Value:

iwpriv ra0 set BADecline=0

0: disable
1: enable

3.1.61 HtProtect

Description: Enable or disable HT protect
Value:

iwpriv ra0 set HtProtect=0

0: disable
1: enable

3.1.62 HtMimoPs

Description: Enable or disable HT MIMO Power saving mode
Value:

iwpriv ra0 set HtMimoPs=0

0: disable
1: enable

3.1.63 HtDisallowTKIP

Description: Enable or disable 11N rate with 11N AP when cipher is TKIP or WEP
Value:

iwpriv ra0 set HtDisallowTKIP=0

0: disable
1: enable

3.1.64 AP2040Rescan

Description: Trigger HT20/40 coexistence to rescan
Value:

iwpriv ra0 set AP2040Rescan=1

1: trigger to rescan

3.1.65 HtBssCoex

Description: Enable or disable HT BSS coexistence
Value:

iwpriv ra0 set HtBssCoex=0

0: disable
1: enable

3.1.66 HtTxStream

Description: Set the number of spatial streams for transmission
Value:

iwpriv ra0 set HtTxStream=1 or 2 or 3

1~3: valid spatial streams

3.1.67 HtRxStream

Description: Set the number of spatial streams for reception
Value:

iwpriv ra0 set HtRxStream=1 or 2 or 3

1~3: valid spatial streams

3.1.68 BASetup

Description: Add an Originator BA entry into the BA table manually.
Value:

iwpriv ra0 set BASetup=00:0c:43:01:02:03-1

→The six 2 digit hex-decimal number(xx) previous are the Mac address,
→The seventh decimal number(d) is the tid value.

3.1.69 BAOriTearDown

Description: Remove an Originator BA entry from the BA table manually.
Value:

iwpriv ra0 set BAOriTearDown=00:0c:43:01:02:03-1

→The six 2 digit hex-decimal number(xx) previous are the Mac address,
→The seventh decimal number(d) is the tid value.

3.1.70 BARecTearDown

Description: Remove an Recipient BA entry from the BA table manually.
Value:

iwpriv ra0 set BARecTearDown=00:0c:43:01:02:03-1

→The six 2 digit hex-decimal number(xx) previous are the Mac address,
→The seventh decimal number(d) is the tid value.

3.1.71 PktAggregate

Description: Enable or disable 11B/G packet aggregation
Value:

iwpriv ra0 set PktAggregate=1

0: disable
1: enable

3.1.72 IEEE80211H

Description: Enable or disable IEEE 802.11h function. Spectrum management.
This field can only be enabled in A band.
Value:

iwpriv ra0 set IEEE80211H=0

0: disable
1: enable

3.1.73 KickStaRssiLow

Description: Set the lowest limitation for AP kicking out STA.
Value:

iwpriv ra0 set KickStaRssiLow=0

0: Disable
0 ~ -100

3.1.74 AssocReqRssiThres

Description: Set AssocReq RSSI Threshold to reject STA with weak signal
Value:

Iwpriv ra0 set AssocReqRssiThres=0

0: Disable
0~ -100

3.2 iNIC specific

3.2.1 QAEnable

Description: enable or disable QA test tool function.

iwpriv ra0 set QAEnable=1

0: disable
1: enable

3.2.2 Console

Description: redirect console information to host.

iwpriv ra0 set Console=1

0: disable
1: enable

3.2.3 EfuseUploadToHost

Description: This command is specific to iNIC solution.
The content of efuse will be uploaded to the iNIC host in iNIC_e2p.bin or iNIC_e2p1.bin .

iwpriv ra0 set EfuseUploadToHost=1

0: disable
1: enable

3.2.4 tpc

Description: Thermal function will be according to criteria with current temperature to configure Ant.
Criteria Value:1~1000
default: 80

Iwpriv ra0 set tpc=80

3.2.5 DfsSwAddCheck

Description: This command is used to add an entry to prevent false detection in specific range.
"ch" is the bbp dfs detection engine ID
"T_Low" is the Radar Period low boundary to filter out.
"T_High" is the Radar Period high boundary to filter out.
"W_Low" is the Radar Width low boundary to filter out.
"W_High" is the Radar Width high boundary to filter out.

iwpriv ra0 set DfsSwAddCheck=ch:T_Low:T_High:W_Low:W_high

Example:
iwpriv ra0 set DfsSwAddCheck=0:100:200:50:500

3.2.6 DfsSwDelCheck

Description: This command is used to delete an entry which was added to filter out radar in specific range.

iwpriv ra0 set DfsSwDelCheck=ch:T_Low:T_High:W_Low:W_high

Example:
iwpriv ra0 set DfsSwDelCheck=0:100:200:50:500

4 Other iwpriv Command

4.1 stat

Description: Show WLAN statistics
Value:

iwpriv ra0 stat

Note:
You can use “iwpriv ra0 set ResetCounter=1” to reset statistics
Also, you can use the following command line shell script to get per-second statistics.
while [1]; do iwpriv ra0 set ResetCounter=1; sleep 1; iwpriv ra0 stat; done;

4.2 get_site_survey

Description: Show site survey result
Value:

iwpriv ra0 get_site_survey

Note: You need to use “iwpriv ra0 set SiteSurvey=” to collect information first

4.3 get_mac_table

Description: Show MAC addresses of connected stations
Value:

iwpriv ra0 get_mac_table

4.4 get_ba_table

Description: Show raw data of the BlockAck table
Value:

iwpriv ra0 get_ba_table

4.5 get_wsc_profile

Description: Show WPS profile information
Value:

iwpriv ra0 get_wsc_profile

4.6 e2p

Description: Read/Write EEPROM content
Value:

```
// Read
iwpriv ra0 e2p offset
// Write
iwpriv ra0 e2p offset=value
```

Note:

offset = hexidecimal address
value = hexidecimal value

4.7 show

You could use iwpriv ra0 show command to display general or specific information. As to specific information, you have to turn on the corresponding function in driver config.

[Format]
iwpriv ra0 show [parameter]

[Parameter list]

1. driverinfo - show driver version
2. stat - show statistics counter
3. stainfo - show MAC address of associated STAs
4. stacountinfo - show TRx byte count of associated STAs
5. stasecinfo - show security information of associated STAs
6. bainfo - show BlockAck information
7. connStatus - show AP-Client connection status
8. reptinfo - show MAC Repeater information
9. wdsinfo - show WDS link list
10. igmpinfo - show all entries in the IGMP Snooping Table
11. mbss - show MBSS PHY mode information
12. blockch - show DFS blocked channel list

[Example]
iwpriv ra0 show driverinfo
Driver version: 2.7.1.6

5 TBD

To Be Defined.

6 WPS

Wi-Fi Protected Setup (WPS) also known as Wi-Fi Simple Config (WSC)

Simple Config Architectural Overview

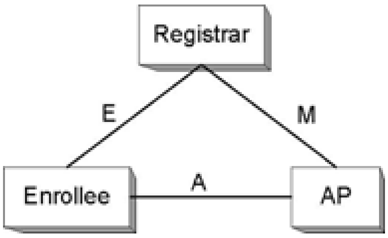
This section presents a high-level description of the Simple Config architecture. Much of the material is taken directly from the Simple Config specification.

Figure 1 depicts the major components and their interfaces as defined by Wi-Fi Simple Config Spec. There are three logical components involved: the Registrar, the access point (AP), and the Enrollee.

The **Enrollee** is a device seeking to join a WLAN domain. Once an Enrollee obtains a valid credential, it becomes a member.

A **Registrar** is an entity with the authority to issue and revoke domain credentials. A registrar can be integrated into an AP.

The **AP** can be either a WLAN AP or a wireless router.



Registration initiation is ordinarily accomplished by a user action such as powering up the Enrollee and, optionally, running a setup wizard on the Registrar (PC).

Note: The WLAN driver needs to set HAS_WSC=y in order to enable WPS functions.

6.1 WPS Profile settings

6.1.1 WscConfMode

Description: Configure WPS role (bitwise OR)

Value:

WscConfMode=7

b'000: 0 Disable
b'001: 1 Enrollee
b'010: 2 Proxy
b'100: 4 Registrar

6.1.2 WscConfStatus

Description: Configure WPS state

Value:

WscConfStatus=1

1: AP is unconfigured
2: AP is configured

6.1.3 WscConfMethods

Description: Setup the configuration methods which Enrollee or Registrar supports
Value:

WscConfMethods=238c

Note:

Hexadecimal value only.
// Bitwise OR all values which DUT supports
0x238c = 0x2008 + 0x0280 + 0x0100 + 0x0004
Virtual Display PIN + Virtual Push Button + Keypad + Label PIN

Config Method	Value
Label PIN	0x0004
External NFC Token	0x0010
Integrated NFC Token	0x0020
NFC Interface	0x0040
Keypad	0x0100
Virtual Push Button	0x0280
Physical Push Button	0x0480
Virtual Display PIN	0x2008
Physical Display PIN	0x4008

6.1.4 WscKeyASCII

Description: Define WPS WPAPSK format and key length for un-configured internal WPS Registrar AP
Value:

WscKeyASCII=0

0: Hex (64-bytes)
1: ASCII (Random length)
8 ~ 63: ASCII length

6.1.5 WscSecurityMode

Description: Define WPS Regitrar's unconfiguraed -> configuraed security mode.
Value:

WscSecurityMode=0

0: WPA2PSK	AES
1: WPA2PSK	TKIP
2: WPAPSK	AES
3: WPAPSK	TKIP

6.1.6 WscDefaultSSID0

Description: Default WPS SSID for AP. After WPS process completes with Enrollee when AP acts as un-configured Registrar, AP will use this SSID as new SSID.

Value:

WscDefaultSSID0=SSID

1~32 characters

6.1.7 WscV2Support

Description: Enable or disable WPS v2.0 support

Value:

WscV2Support=1

0: disable

1: enable

6.2 WPS iwpriv command

6.2.1 WscConfMode

Description: Configure WPS role (bitwise OR)

Value:

iwpriv ra0 set WscConfMode=7

b’000: 0 Disable

b’001: 1 Enrollee

b’010: 2 Proxy

b’100: 4 Registrar

6.2.2 WscConfStatus

Description: Configure WPS state

Value:

iwpriv ra0 set WscConfStatus=1

1: AP is unconfigured

2: AP is configured

6.2.3 WscMode

Description: Configure WPS mode

Value:

iwpriv ra0 set WscMode=1

1: PIN Mode

2: PBC Mode

6.2.4 WscStatus

Description: Get WPS Configured Methods.

Value:

iwpriv ra0 set WscStatus=0

0:	Not Used
1:	Idle
2:	WSC Process Fail
3:	Start WSC Process
4:	Received EAPOL-Start
5:	Sending EAP-Req(ID)
6:	Receive EAP-Rsp(ID)
7:	Receive EAP-Req with wrong WSC SMI Vendor Id
8:	Receive EAPReq with wrong WSC Vendor Type
9:	Sending EAP-Req(WSC_START)
10:	Send M1
11:	Received M1
12:	Send M2
13:	Received M2
14:	Received M2D
15:	Send M3
16:	Received M3
17:	Send M4
18:	Received M4
19:	Send M5
20:	Received M5
21:	Send M6
22:	Received M6
23:	Send M7
24:	Received M7
25:	Send M8
26:	Received M8
27:	Processing EAP Response (ACK)
28:	Processing EAP Request (Done)
29:	Processing EAP Response (Done)
30:	Sending EAP-Fail
31:	WSC_ERROR_HASH_FAIL
32:	WSC_ERROR_HMAC_FAIL

33: WSC_ERROR_DEV_PWD_AUTH_FAIL
34: Configured

6.2.5 WscPinCode

Description: Input Enrollee's Pin Code to AP-Registrar.
Value:

iwpriv ra0 WscPinCode xxxxxxxx

xxxxxxx = {00000000 ~ 99999999}

6.2.6 WscOOB

Description: Reset WPS AP to the OOB (out-of-box) configuration.
Value:

iwpriv ra0 set WscOOB=1

0: disable
1: enable

6.2.7 WscGetConf

Description: Trigger WPS AP to do simple config with WPS Client.
Value:

iwpriv ra0 set WscGetConf=1

0: disable
1: enable

6.2.8 WscGenPinCode

Description: Randomly generate enrollee PIN code
Value:

iwpriv ra0 set WscGenPinCode=1

1

6.2.9 WscVendorPinCode

Description: Input vendor's Pin Code to AP-Registrar.
Value:

iwpriv ra0 set WscVendorPinCode=xxxxxxxx

xxxxxxxx: 8 digit pin code

6.2.10 WscSecurityMode

Description: Set WPS regitrar's unconfiguraed -> configuraed security mode.

Value:

iwpriv ra0 set WscSecurityMode=0

0 : WPA2PSK AES
1 : WPA2PSK TKIP
2 : WPAPSK AES
3 : WPAPSK TKIP

6.2.11 WscMultiByteCheck

Description: Set multi byte check is enabled or disabled.
Value:

iwpriv ra0 set WscMultiByteCheck=1

0: disable
1: enable

6.2.12 WscVersion

Description: Set WPS support version
Value:

iwpriv ra0 set WscVersion=10

xx: Hex value

6.2.13 WscVersion2

Description: Set WPS version of V2 support
Value:

iwpriv ra0 set WscVersion2=10

xx: Hex Value

6.2.14 WscV2Support

Description: enable or disable WPS V2.0 support
Value:

iwpriv ra0 WscV2Support=1

0: disable
1: enable

6.2.15 WscFragment

Description: enable or disable WPS fragment
Value:

iwpriv ra0 WscFragment=0

0: disable
1: enable

6.2.16 WscFragmentSize

Description: Set the size of WPS fragmentation.
Value:

iwpriv ra0 set WscFragmentSize=128

128~300

6.2.17 WscSetupLock

Description: enable or disable WPS setup lock
Value:

iwpriv ra0 set WscSetupLock=1

0: disable
1: enable

6.2.18 WscSetupLockTime

Description: Configure WPS setup lock time
Value:

iwpriv ra0 set WscSetupLockTime=0

0: lock forever
Unit: minute

6.2.19 WscMaxPinAttack

Description: Configure WPS pin attack Max time.
Value:

iwpirv ra0 set WscMaxPinAttack

0:Disable
1-10

6.2.20 WscExtraTlvTag

Description: Add extra TLV tag to Beacon, probe response and WSC EAP messages
Value:

iwpriv ra0 set WscExtraTlvTag=1088

Hex value: 0000 ~ FFFF
Example: 1088

6.2.21 WscExtraTlvType

Description: Define data format of extra TLV value
Value:

iwpriv ra0 set WscExtraTlvType=1

0: ASCII string
1: Hex string

6.2.22 WscExtraTlvData

Description: Add extra TLV data to Beacon, probe response and WSC EAP messages
Value:

iwpriv ra0 set WscExtraTlvData=

ASCII string or Hex string

6.2.23 WscStop

Description: Stop WPS process.
Value:

iwpriv ra0 set WscStop

6.2.24 WPS iwpriv command example

6.2.24.1 Disable WPS support

iwpriv ra0 set WscConfMode=0

6.2.24.2 Enable WPS Function

iwpriv ra0 set WscConfMode =7 (Binary: 111)
(AP could be Registrar(0x4), Proxy(0x2) or Enrollee(0x1))

6.2.24.3 WPS AP SC (Simple Config) State

iwpriv ra0 set WscConfStatus=1 (AP is un-configured)
iwpriv ra0 set WscConfStatus=2 (AP is configured)

6.2.24.4 WPS Configured Methods

iwpriv ra0 set WscMode =1 (use PIN code)
iwpriv ra0 set WscMode =2 (use PBC)

6.2.24.5 Input Enrollee's Pin Code to AP-Registrar

iwpriv ra0 set WscPinCode=xxxxxxxx

6.2.24.6 Reset WPS AP to the OOB configuration

iwpriv ra0 set WscOOB=1
(Security: WPAPSK/TKIP, psk: "RalinkInitialAPxx1234" ; SC state: 0x1)

(SSID: RalinkInitialAPxxxxxx, last three characters of AP MAC address)

6.2.24.7 Trigger WPS AP to do simple config with WPS Client

iwpriv ra0 set WscGetConf=1

6.2.24.8 AP services as Enrollee by using PIN code

iwpriv ra0 set WscMode=1
iwpriv ra0 set WscGetConf=1

6.2.24.9 AP services as Enrollee by using PBC

iwpriv ra0 set WscMode=2
iwpriv ra0 set WscGetConf=1

6.2.24.10 AP services as Internal Registrar using PIN code

iwpriv ra0 set WscMode=1
iwpriv ra0 set WscPinCode=xxxxxxxx (PIN code from Enrollee, len=8)
iwpriv ra0 set WscGetConf=1

6.2.24.11 AP services as Internal Registrar using PBC

iwpriv ra0 set WscMode=2
iwpriv ra0 set WscGetConf=1

6.2.24.12 Get WPS Profile from external registrar

iwpriv ra0 get_wsc_profile

6.3 WPS AP Setup Procedure

To run the Access Point (as Enrollee or with Registrar capabilities).
The following scenarios are currently supported:

1. Initial Access Point (AP) setup, with the Registrar configuring the Access Point
   - 1.1. One WiFi-enabled laptop is setup as the AP acting as an Enrollee
   - 1.2. Another WiFi-enabled laptop is setup as a station acting as the Registrar
   - 1.3. Two sub cases are 1a) using EAP transport and 1b) using UPnP transport

2. Configuration of a WiFi client, using an AP with a built-in registrar
   - 2.1. One WiFi-enabled laptop is setup as the AP with registrar functionality Another WiFi-enabled laptop is setup as a station acting as an Enrollee

3. Configuration of a WiFi client using an external registrar. AP acts as a proxy and communicates with the client over EAP and with the Registrar over UPnP.
   - 3.1. One WiFi-enabled laptop is setup as a station acting as an Enrollee
   - 3.2. Second WiFi-enabled laptop is setup as the AP with proxy functionality
   - 3.3. Third laptop is setup as the registrar. The registrar and the AP are connected over Ethernet.

6.3.1 Running the WPS command-line application

Run the protocol from the console.
First, run UPNP deamon like below:
wscd -w /etc/xml -m 1 -d 3 & (if your xml file in /etc/xml)

use iwpriv command trigger wps, like below:

iwpriv ra0 set WscConfMode=7
iwpriv ra0 set WscConfStatus=1
iwpriv ra0 set WscMode=1
iwpriv ra0 set WscPinCode=31668576
iwpriv ra0 set WscGetConf=1
iwpriv ra0 set WscStatus=0

1. AP services as Enrollee:
   1.1. If AP-Enrollee SC state is 0x1, AP will restart with new configurations.
   1.2. If AP-Enrollee SC state is 0x2, AP sends own configurations to external-registrar and ignores configurations from external-registrar.

2. AP services as Registrar:
   2.1. If AP-Registrar SC state is 0x1, the security mode will be WPAPSK/TKIP and generate random 64bytes psk; after process, AP will restart with new security.

3. WPS AP only services one WPS client at a time.
   3.1. WPS AP only can work in ra0.
   3.2. After WPS configuration finishes, Ralink AP driver writes new configuration to Cfg structure and DAT file.

4. Write items to MBSSID Cfg structure are as below:
   4.1. Ssid
   4.2. AuthMode
   4.3. WepStatus
   4.4. PMK
   4.5. DefaultKeyId.

5. Write items to SharedKey table are as below:
   5.1. Key
   5.2. CipherAlg

6. Write items to DAT file are as belw:
   6.1. SSID
   6.2. AuthMode
   6.3. EncrypType
   6.4. WPAPSK
   6.5. WscConfStatus
   6.6. DefaultKeyID

Note: wscd daemon must be ported to the target platform first.

6.3.2 Initial AP setup with Registrar Configuring AP (EAP/UPnP)

To run command-line console in this mode do:

[Unconfigured AP] ← EAP/UPnP → [Registrar]

Note:

Please make sure upnp deamon is running. After the success of WPS registration, Configured AP will act as a proxy forward EAP and Upnp.)

1. **PIN**

(1) **on AP side**

- iwpriv ra0 set WscConfMode=7
- iwpriv ra0 set WscConfStatus=1
- iwpriv ra0 set WscMode=1
- iwpriv ra0 set WscGetConf=1

(2) **on Registrar side**

- When prompted for the enrollee's PIN, Enter the AP's PIN. Enter the new SSID and new Security for the AP when prompted.
- The registration process will start, and the application will display the result of the process on completion.

2. **PBC**

(1) **on AP side**

- iwpriv ra0 set WscConfMode=7
- iwpriv ra0 set WscConfStatus=1
- iwpriv ra0 set WscMode=2
- iwpriv ra0 set WscGetConf=1

(2) **on Registrar side**

- Select push-button".
- The registration process will start, and the application will display the result of the process on completion.

The security config will be written out to the AP and registrar config files.

6.3.3 Adding an Enrollee to AP+Registrar (EAP)

To run command-line console in this mode do:

[AP+Registrar] ← EAP → [Client]

Note:

Please make sure WPS AP configure status is configured, if AP is un-configure, when WPS AP configure client, it will change configure status to configured and auth mode are WPA-PSK)

1. **PIN**

(1) **on AP side**

- iwpriv ra0 set WscConfMode=7
- iwpriv ra0 set PinCode=31668576 (enter the enrollee's PIN, the PIN from WPS client)
- iwpriv ra0 set WscMode=1
- iwpriv ra0 set WscGetConf=1.
- The registration process will begin, and the console will display the result of the process on completion.

(2) **on Client (Enrollee) side**

- Select PIN process.
- The process will start, and the application will display the result of the process on completion

2. **PBC**

(1) **on AP side**

- iwpriv ra0 set WscConfMode=7
- iwpriv ra0 set WscMode=2
- iwpriv ra0 set WscGetConf=1.
- The registration process will start, and the application will display the result of the process on completion.

(2) **on Client (Enrollee) side**

- Select PBC process.

- The process will start, and the application will display the result of the process on completion

If the registration is successful, on the client will be re-configured with the new parameters, and will connect to the AP with these new parameters.

6.3.4 Adding an Enrollee with Eternal Registrar (UPnP/EAP)

To run command-line console in this mode do:

[Registrar] ← PnP → [AP] ← EAP → [Client]

1. **PIN**
   (1) on Registrar side
   - When prompted for the enrollee's PIN, Enter the enrollee's PIN.
   - AP Nothing to be selected..
   - The registration process will begin, and the application will display the result of the process on completion.

   (2) on Client (Enrollee) side
   - Select PIN process
   - The process will start, and the application will display the result of the process on completion
2. **PBC**
   (1) on Registrar side
   - Select "push-button".
   - AP Nothing to be selected.
   - The registration process will begin, and the application will display the result of the process on completion.

   (2) on Client (Enrollee) side
   - Select PBC process
   - The registration process will start, and the application will display the result of the process on completion.

6.3.5 WPS Config status

6.3.5.1 Over View

The 'Simple Config State' of WPS attribute in WPS IEs contained in beacon and probe response indicates if a device is configured.If an AP is shipped from the factory in the Not-Configured state (Simple Config State set to 0x01), then the AP must transition to the Configured state (Simple Config State set to 0x02) if any of the following occur:

1. Configuration by an external registrar.

The AP sends the WSC_Done message in the External Registrar configuration process.

2. Automatic configuration by internal registrar.

The AP receives the WSC_Done response in the Enrollee Registration Process from the first Enrollee.
Note:
The internal registrar waits until successful completion of the protocol before applying the automatically generated credentials to avoid an accidental transition from unconfigured to configured in the case that a neighbouring device tries to run WSC before the real enrollee, but fails. A failed attempt does not change the configuration of the AP, nor the Simple Config State.

3. Manual configuration by user.

A user manually configures the AP using whatever interface(s) it provides to modify any one of the following:

- the SSID
- the encryption algorithm
- the authentication algorithm
- any key or pass phrase

If the AP is shipped from the factory in the Not Configured state (Simple Config State set to 0x01), then a factory reset must revert the Simple Config State to Not Configured.

If the AP is shipped from the factory pre-configured with WPA2-Personal mixed mode and a randomly generated key, the Simple Config State may be set to 'Configured' (0x2) to prevent an external registrar from overwriting the factory settings. A factory reset must restore the unit to the same configuration as when it was shipped.

6.4 Basic operation of Ralink WPS AP

6.4.1 Configure APUT using PIN method through a WLAN external Registrar

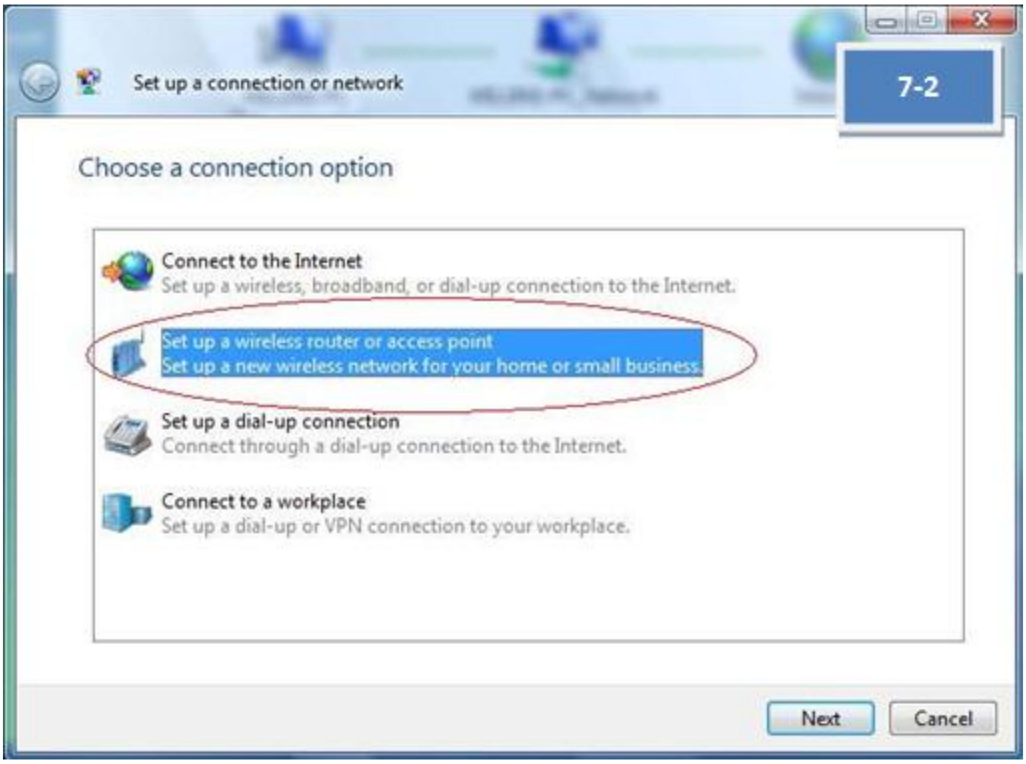
1. [Ralink AP] - Turn on the Ralink AP
2. [Ralink AP] - To change AP ability "iwpriv ra0 set WscConfMode=7"
3. [Ralink AP] - To change from configured to un-configured state: "iwpriv ra0 set WscConfStatus=1 "
4. [Ralink AP] - To change config method to PIN "iwpriv ra0 set WscMode=1"
5. [Ralink AP] - Trigger Ralink AP start process WPS protocol "iwpriv ra0 set WscGetConf=1"
6. [Intel WPS STA] - The Registrar on Intel STA will be configured with the new parameters (SSID = "scaptest4.1.2ssid" and WPA(2)-PSK="scaptest4.1.2psk") which should be entered when prompted
7. [Intel WPS STA] - Read AP's PIN from console and enter the PIN at Intel STA.
8. [Intel WPS STA] - Verify that Intel STA successes to ping to Ralink AP
9. [Ralink STA] - Manually configure Ralink STA with the new parameters (SSID = "scaptest4.1.2ssid" and WPA (2)-PSK = "scaptest4.1.2psk").
10. [Intel WPS STA] - Verify that Intel STA successes to ping to Ralink STA

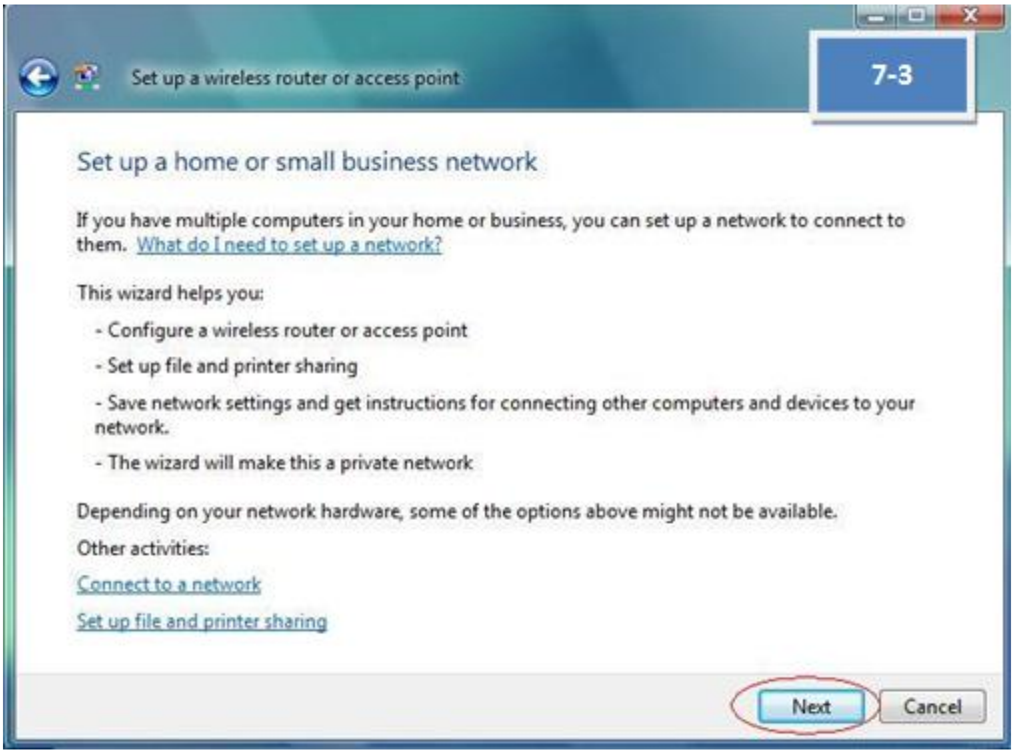
6.4.2 Configure APUT using PIN method through a wired external registrar

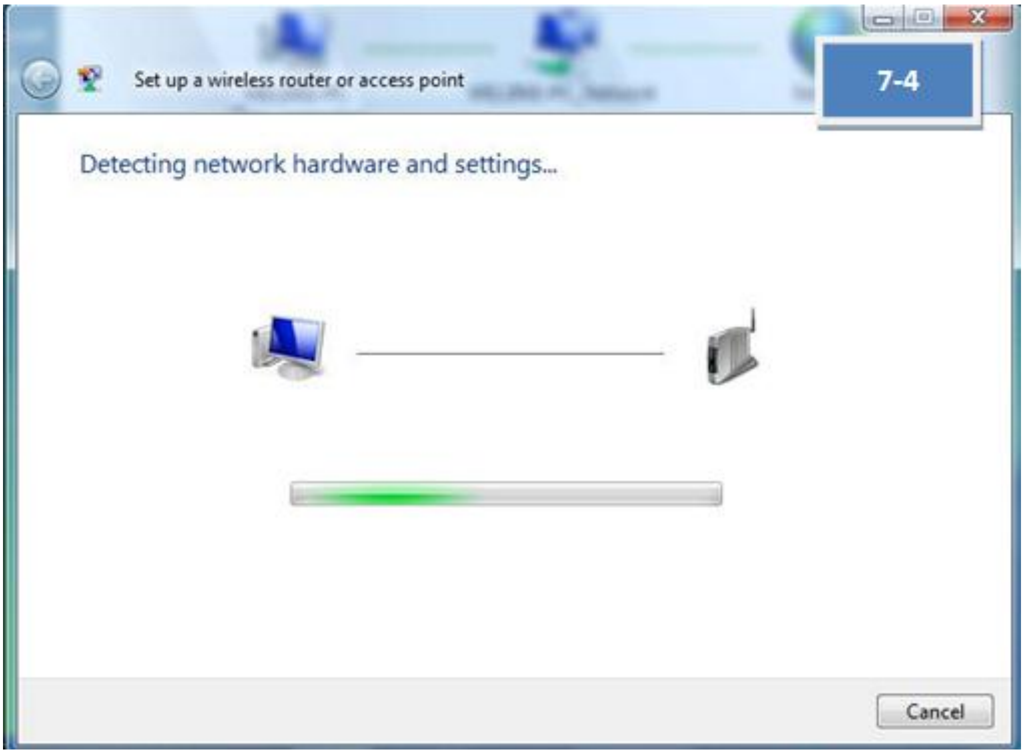
1. [Ralink AP] - Turn on the Ralink AP
2. [Ralink AP] - Connect the Ethernet cable between AP and extern registrar(Windows Vista) and make sure you can pin our device from extern registrar first!
3. [Ralink AP] - To change AP ability "iwpriv ra0 set WscConfMode=7"
4. [Ralink AP] - To change from configured to un-configured state: "iwpriv ra0 set WscConfStatus=1 "
5. [Ralink AP] - To change config method to PIN "iwpriv ra0 set WscMode=1"
6. [Ralink AP] - Trigger Ralink AP start process WPS protocol "iwpriv ra0 set WscGetConf=1"
7. [Microsoft STA] - The Registrar on Microsoft STA will be configured with the new wireless configuration settings (SSID = "scaptest4.1.3ssid" and WPA (2)-PSK="scaptest4.1.3psk"), which should be entered when prompted.

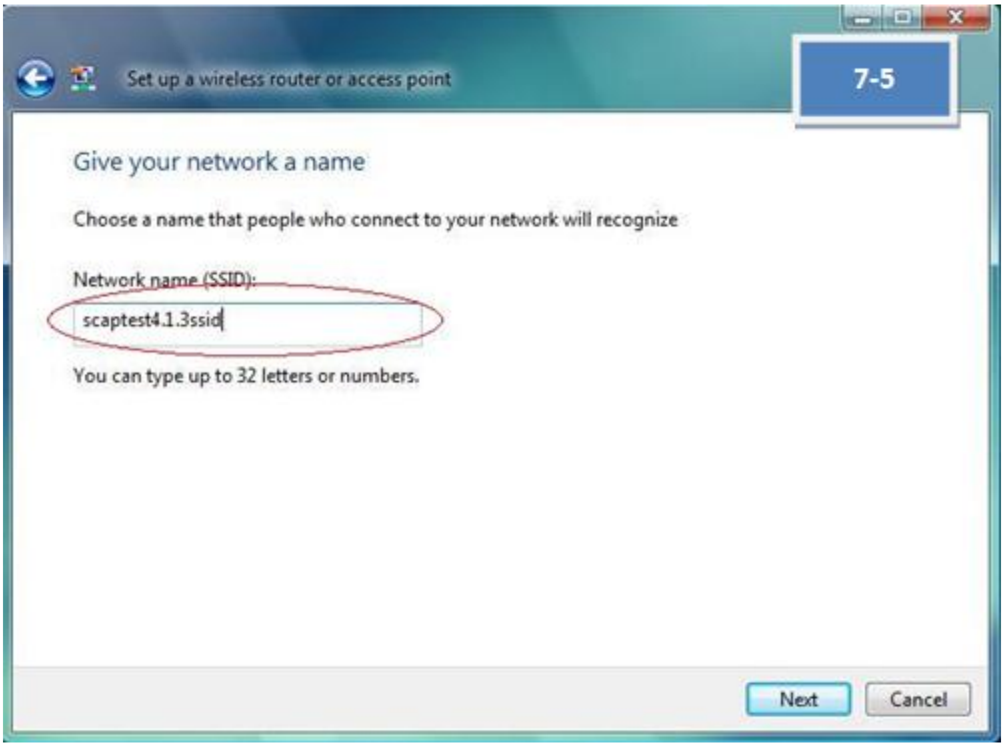
Please refer to below figures [7-1] to [7-6].


Control Panel ▸ Network and Sharing Center
Search
Tasks
View computers and devices
Connect to a network
Set up a connection or network
Manage network connections
Diagnose and repair
Network and Sharing Center
7-1
WILLING-PC
(This computer)
WILLING-PC_Network
Internet
WILLING-PC_Network (Private network)
Customize
Access
Local and Internet
Connection
Local Area Connection
View status
Sharing and Discovery
Network discovery
On
File sharing
On
Public folder sharing
Off
Printer sharing
Off (no printers installed)
Password protected sharing
On
Media sharing
Off
See also
Internet Options
Windows Firewall
Show me all the files and folders I am sharing
Show me all the shared network folders on this computer


Set up a connection or network
7-2
Choose a connection option
Connect to the Internet
Set up a wireless, broadband, or dial-up connection to the Internet.
Set up a wireless router or access point
Set up a new wireless network for your home or small business.
Set up a dial-up connection
Connect through a dial-up connection to the Internet.
Connect to a workplace
Set up a dial-up or VPN connection to your workplace.
Next
Cancel


Set up a wireless router or access point
7-3
Set up a home or small business network
If you have multiple computers in your home or business, you can set up a network to connect to them. What do I need to set up a network?
This wizard helps you:
- Configure a wireless router or access point
- Set up file and printer sharing
- Save network settings and get instructions for connecting other computers and devices to your network.
- The wizard will make this a private network
Depending on your network hardware, some of the options above might not be available.
Other activities:
Connect to a network
Set up file and printer sharing
Next
Cancel

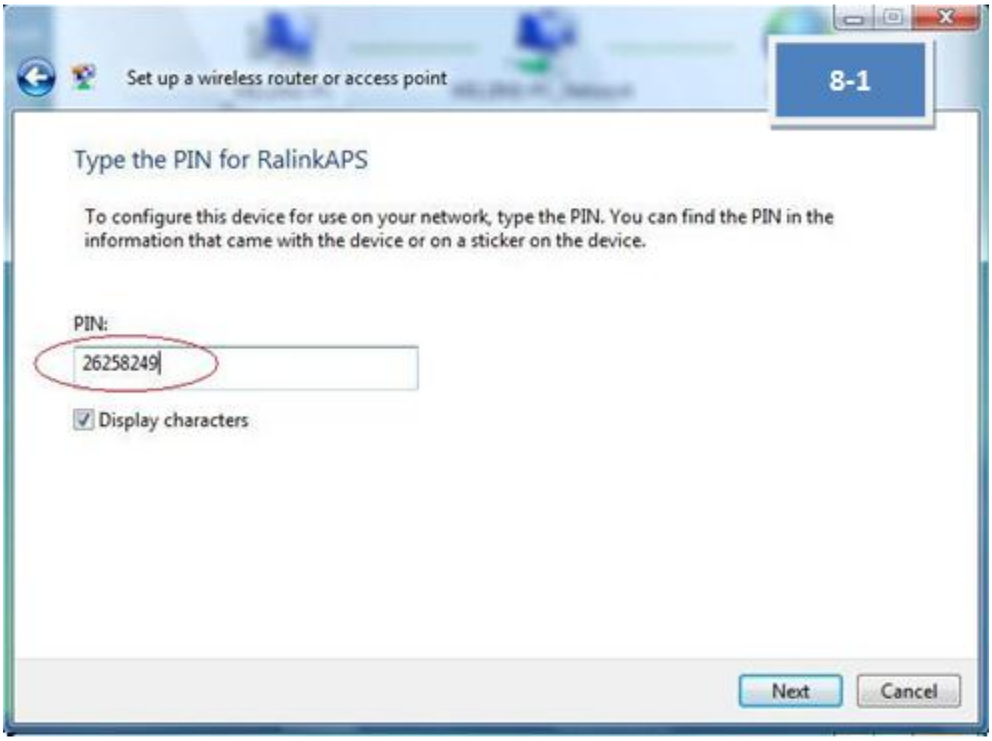

Set up a wireless router or access point
7-4
Detecting network hardware and settings...
Cancel

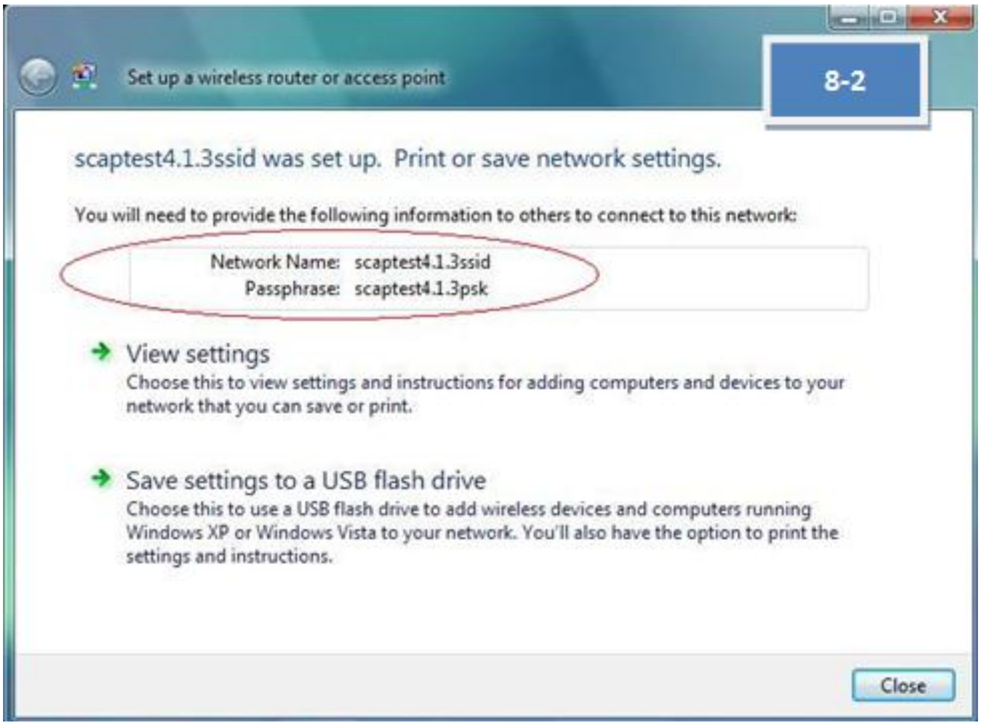




1. [Microsoft STA] - Read AP's PIN from console and enter the PIN at Microsoft STA.

Please refer to below figures [8-1] to [8-2].





1. [Ralink STA] - Manually configure Ralink STA with the new parameters (SSID = “scaptest4.1.3ssid” and WPA (2)-PSK passphrase= “scaptest4.1.3psk”).
2. [Ralink STA] - Verify that Ralink STA successes to ping to Microsoft STA.

6.4.3 Add devices using external Registrars

1. [Ralink AP] - Turn on the APUT.
2. [Ralink STA] - Turn on the Ralink STA.

3. [Ralink STA] - Push PIN button.
4. [Microsoft STA] - Search will be configure enrollee (you can in control->network and internet->network and sharing center->add a device to the network). Enter the enrollee's PIN(Ralink STA) at Microsoft STA when prompted.
5. [Ralink AP] - Do not thing.
6. [Ralink STA] - Verify that Ralink STA successes to ping Ralink A.

6.4.4 How to know WPS AP services as Internal Registrar, Enrollee or Proxy

It depends on the content of EAP-Response/Identity from WPS Client.

- ⇨ When identity is "WFA-SimpleConfig-Registrar-1-0":
  WPS AP would service as Enrollee. (After set trigger command)
- ⇨ When identity is "WFA-SimpleConfig-Enrollee-1-0":
  WPS AP would service as Internal Registrar and Proxy.
  Without trigger command, WPS AP services as proxy only.

6.4.5 How to know WPS AP PinCode

Use ioctl query **RT_OID_WSC_PIN_CODE** OID to get AP PinCode.

6.4.6 Notes for WPS

1. AP services as Enrollee:
   1.1. If AP-Enrollee SC state is 0x1, AP's configuration is changeable and will restart with new configurations.
   1.2. If AP-Enrollee SC state is 0x2, AP's configuration is un-changeable. AP sends own configurations to external-registrar and ignores configurations from external-registrar.
2. AP services as Registrar:
   2.1. If AP-Registrar SC state is 0x1, the security mode will be WPAPSK/TKIP and generate random 64bytes psk; after process, AP will restart with new security.
3. AP services as Proxy:
   3.1. The value of SC state has no effect in proxy mode.
   3.2. WPS AP only services one WPS client at a time.
   3.3. WPS AP only can work in ra0.

6.4.7 Compile flag for WPS AP

WFLAGS += -DWSC_SUPPORT

6.4.8 WPS related Document

1. Wi-Fi Protected Setup Specification v1.0 (member only)
2. Wi-Fi Protected Setup White Paper
3. Introducing Wi-Fi Protected Setup
4. WSC Linux* Reference Implementation
5. How to Use Windows Connect Now Configuration to Enable Simple Setup for Consumer Wi-Fi Networks [WinHEC 2006; 5.83 MB]
6. Network Infrastructure Device Implementer's Guide

6.5 UPNP Daemon HOWTO

6.5.1 Build WPS UPnP Daemon

Requirements:

1. Linux platform
2. Ralink wireless driver version which support WPS
3. Libupnp
   - ➩ You can download the libupnp source code from the following URL: http://upnp.sourceforge.net/
   - ➩ libupnp-1.3.1 is preferred version. For other versions, you may need to patch our modification to the library yourself.
4. POSIX thread library
   - ➩ Both libupnp and our WPS UPnP daemon need the POSIX thread library, following are recommended pthread library version.
     - For uCLibc, need the version >= 0.9.27
     - For GLIBC, need the version >= 2.3.2
   - ➩ If your pthread library is older than upper list, you may need to upgrade it.

Build and Run:

1. Modify the "$(work_directory)/wsc_upnp/Makefile" and change the compile flags depends on your target platform.
   - ➩ Ex. For arm-Linux target platform, you may need to set the following fags:
     - CROSS_COMPILE = arm-Linux-
     - TARGET_HOST = arm-Linux
     - **WIRELESS_H_INCLUDE_PATH = /usr/src/kernels/2.6.11-1.1369_FC4-smp-i686/include/**
2. Modify the "$(work_directory)/wsc_upnp/libupnp-1.3.1/Makefile.src" and change the configure parameters.
   - ➩ Ex. For big-endian system, you may need to add CFAGS as following:
     - ./configure --host=$(TARGET_HOST) CFLAGS="-mbig-endian"
3. Compile it
   - ➩ Run "make" in "$(work_directory)/wsc_upnp", after successful compilation, you will get an execution file named "wscd".
4. Install
   - ➩ Create a sub-directory named "xml" in the "/etc" of your target platform
   - ➩ Copy all files inside in "$(work_directory)/wsc_upnp/xml" to "/etc/xml"
     - Copy the "wscd" to the target platform.
5. Run it
   - ➩ Before run it, be sure the target platform already **has set the default route or has a route entry for subnet 239.0.0.0 (For UuPnP Multicast)**. Or the WPS daemon will failed when do initialization.
   - ➩ Now you can run it by following command:
     - /bin/wscd –m 1 –d 3

Related Document:

1. WPS Specification (Simple_Config_v1.0g.pdf)
2. UPnP Device Architecture 1.0
3. Windows Connect Now-NET Version 1.0
4. WFAWLANConfig:1 Service Template Version 1.01
5. WFA Device:1 Device Template Version 1.01

6.6 WPS Command & OID Example

6.6.1 Iwpriv command without argument

iwpriv command:

```
iwpriv ra0 wsc_start
iwpriv ra0 wsc_stop
iwpriv ra0 wsc_gen_pincode
```

OID:

Example:

```
memset(&lwreq, 0, sizeof(lwreq));
sprintf(lwreq.ifr_name, "ra0", 3);
iwreq.u.mode = WSC_STOP;
/* Perform the private ioctl */
if(ioctl(skfd, RTPRIV_IOCTL_SET_WSC_PROFILE_U32_ITEM, &lwreq) < 0)
{
fprintf(stderr, "Interface doesn't accept private ioctl...\n");
```

7 WMM

7.1 Introduction

IEEE 802.11e amendment is to provide basic QoS features to 802.11 network and Wi-Fi Multimedia (WMM) is a WFA interoperability certification based on the IEEE 802.11e standard. WMM prioritizes wireless traffic according to four Access Categories, including Voice (VO), Video (VI), Best Effort (BE) and Background (BK).

7.2 WMM iwpriv command

7.2.1 WmmCapable

Description: Enable or disable WMM QoS function
Value:

iwpriv ra0 set WmmCapable=1

0: disable
1: enable

7.3 Parameters in RT2860AP.dat

7.3.1 WmmCapable

Description: Enable or disable WMM QoS function
Value:

WmmCapable=1

0: disable
1: enable

Note: Only WmmCapable has iwpriv command support

7.3.2 APSDCapable

Description: WMM Automatic Power Save Delivery (APSD) function configuration
Value:

APSDCapable=0

0: disable
1: enable

7.3.3 APAifsn

Description: AP arbitration interframe space number configuration
Value:

APAifsn=3;7;1;1

AC_BE;AC_BK;AC_VI;AC_VO

7.3.4 APCwmin

Description: AP contention window minimum (exponent) configuration
Value:

APCwmin=4;4;3;2

AC_BE;AC_BK;AC_VI;AC_VO

7.3.5 APCwmax

Description: AP contention window maximum (exponent) configuration
Value:

APCwmax=6;10;4;3

AC_BE;AC_BK;AC_VI;AC_VO

7.3.6 APTxop

Description: AP Transmit Opportunity configuration (unit: 32μs)
Value:

APTxop=0;0;94;47

AC_BE;AC_BK;AC_VI;AC_VO

7.3.7 APACM

Description: AP Admission Control Mandatory configuration
Value:

APACM=0;0;0;0

AC_BE;AC_BK;AC_VI;AC_VO

7.3.8 BSSAifsn

Description: STA arbitration interframe space number configuration
Value:

BSSAifsn=3;7;2;2

AC_BE;AC_BK;AC_VI;AC_VO

7.3.9 BSSCwmin

Description: STA contention window minimum (exponent) configuration
Value:

BSSCwmin=4;4;3;2

AC_BE;AC_BK;AC_VI;AC_VO

7.3.10 BSSCwmax

Description: STA contention window maximum (exponent) configuration
Value:

BSSCwmax=10;10;4;3

AC_BE;AC_BK;AC_VI;AC_VO

7.3.11 BSSTxop

Description: STA Transmit Opportunity configuration (unit: 32μs)
Value:

BSSTxop=0;0;94;47

AC_BE;AC_BK;AC_VI;AC_VO

7.3.12 BSSACM

Description: STA Admission Control Mandatory configuration
Value:

BSSACM=0;0;0;0

AC_BE;AC_BK;AC_VI;AC_VO

7.3.13 AckPolicy

Description: Acknowledgement policy configuration
Value:

AckPolicy=0;0;0;0

0: Normal Ack or Implicit Block Ack Request
1: No Ack
2: No explicit acknowledgement
3: Block Ack

AC_BE;AC_BK;AC_VI;AC_VO

7.4 How to Run WMM test

1. WmmCapable=1
2. TxBurst=**0**
3. Parameters for AP

APAifsn=3;7;1;1	// AC_BE;AC_BK;AC_VI;AC_VO
APCwmin=4;4;3;2	// AC_BE;AC_BK;AC_VI;AC_VO
APCwmax=6;10;4;3	// AC_BE;AC_BK;AC_VI;AC_VO
APTxop=0;0;94;47	// AC_BE;AC_BK;AC_VI;AC_VO
APACM=0;0;0;0	// AC_BE;AC_BK;AC_VI;AC_VO

4. Parameters for all STAs

BSSAifsn=3;7;2;2	// AC_BE;AC_BK;AC_VI;AC_VO
BSSCwmin=4;4;3;2	// AC_BE;AC_BK;AC_VI;AC_VO
BSSCwmax=10;10;4;3	// AC_BE;AC_BK;AC_VI;AC_VO
BSSTxop=0;0;94;47	// AC_BE;AC_BK;AC_VI;AC_VO
BSSACM=0;0;0;0	// AC_BE;AC_BK;AC_VI;AC_VO

5. Ack policy

AckPolicy=0;0;0;0	// AC_BE;AC_BK;AC_VI;AC_VO;

All default values comply with the Wi-Fi spec.

8 IEEE802.11d & IEEE802.11h

8.1 IEEE802.11d

Regulatory Domains

To turn on IEEE802.11d, just fill up the parameter of 'CountryCode', according to ISO3166 code list. This parameter can work in A/B/G band.

The parameter of "CountryCode' needs to match with 'CountryRegion' or 'CountryRegionABand' depends on A or B/G band

Wi-Fi test requirement for IEEE802.11d

Country code IE (0x07) includes in beacon frame and probe response

Power constraint IE (32) includes in beacon frame and probe response

8.2 IEEE802.11h

Spectrum and Transmit Power Management

1. To turn on IEEE802.11h, just fill up the parameters of 'IEEE80211H', 'AutoChannelSelect' as 1, WirelessMode set as 3 to support A band. This parameter can work in only A band.
2. Use 'CSPeriod' to determine how many beacons before channel switch
3. Driver will turn off BBP tuning temporarily in radar detection mode
4. If turn on IEEE802.11h, AP will have 60sec to do channel available check, and will not send beacon and can not be connect.
5. Wi-Fi test requirement for IEEE802.11h
   - Force AP switch channel, AP will stop beacon transmit between 15 sec
   - At least five beacon includes channel switch announcement IE (37)in beacon frame
6. ETSI test requirement, please refer to ETSI EN 301 893 for V1.2.3 detail

Table D.1: DFS requirement values

Parameter	Value
Channel Availability Check Time	60 s
Channel Move Time	10 s
Channel Closing Transmission Time	260 ms

Table D.2: Interference Threshold values, Master

Maximum Transmit Power	Value (see note)
≥ 200 mW	-64 dBm
< 200 mW	-62 dBm
NOTE: This is the level at the input of the receiver assuming a 0 dBi receive antenna.	

Table D.3: Interference Threshold values, Slave

Maximum Transmit Power	Value (see note)
≥ 200 mW	-64 dBm
< 200 mW	N/A
NOTE: This is the level at the input of the receiver assuming a 0 dBi receive antenna.	

9 SECURITY

9.1 All possible combinations of security policy

Type I. Without Radius (IEEE8021X has to be **False**)

	OPEN	SHARED	WEPAUTO
NONE	V	X	X
WEP	V	V	V
802.1x daemon	Off	Off	Off

Type II. With Radius (Non-WiFi standard) (IEEE8021X has to be **True**)

	OPEN
NONE	V
WEP	V
802.1x daemon	On

Type III. With WFA WPA/WPA2 (IEEE8021X has to be **False**)

	WPAPSK	WPA2PSK	WPAPSK WPA2PSK	WPA	WPA2	WPA WPA2
TKIP	V	V	V	V	V	V
AES	V	V	V	V	V	V
TKIPAES	V	V	V	V	V	V
802.1x daemon	Off	Off	Off	**On**	**On**	**On**

9.2 Security iwpriv command

9.2.1 AuthMode

Description: WLAN security authentication mode

Value:

iwpriv ra0 set AuthMode=OPEN

OPEN	Open system
SHARED	Shared key system
WEPAUTO	Auto switch between OPEN and SHARED
WPAPSK	WPA Pre-Shared Key (Infra)
WPA2PSK	WPA2 Pre-Shared Key (Infra)
WPAPSKWPA2PSK	WPAPSK/WPA2PSK mixed mode (Infra)
WPA	WPA Enterprise mode (Need wpa_supplicant)
WPA2	WPA2 Enterprise mode (Need wpa_supplicant)
WPA1WPA2	WPA/WPA2 mixed mode (Need wpa_supplicant)

9.2.2 EncrypType

Description: WLAN security encryption type
Value:

iwpriv ra0 set EncrypType=NONE

NONE	No encryption
WEP	Wired Equivalent Privacy
TKIP	Temporal Key Integrity Protocol
AES	Advanced Encryption Standard
TKIPAES	Mixed cipher

9.2.3 DefaultKeyID

Description: Default key ID (WEP only)
Value:

iwpriv ra0 set DefaultKeyID=1

The ID range is 1~4

9.2.4 Key1

Description: Key 1 string (WEP only)
Value:

iwpriv ra0 set Key1=aaaaa

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.2.5 Key2

Description: Key 2 string (WEP only)
Value:

iwpriv ra0 set Key2=aaaaa

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.2.6 Key3

Description: Key 3 string (WEP only)
Value:

iwpriv ra0 set Key3=aaaaa

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.2.7 Key4

Description: Key 4 string (WEP only)
Value:

iwpriv ra0 set Key4=aaaaa

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.2.8 WPAPSK

Description: WLAN security password for TKIP/AES
Value:

iwpriv ra0 set WPAPSK=12345678

8~63 ASCII characters
64 hexadecimal characters

9.2.9 WpaMixPairCipher

Description: Providing more flexible combination of cipher suite
Value:

iwpriv ra0 set WpaMixPairCipher=WPA_TKIP_WPA2_AES

WPA_AES_WPA2_TKIPAES
WPA_AES_WPA2_TKIP
WPA_TKIP_WPA2_AES
WPA_TKIP_WPA2_TKIPAES
WPA_TKIPAES_WPA2_AES
WPA_TKIPAES_WPA2_TKIPAES
WPA_TKIPAES_WPA2_TKIP

9.3 Parameters in RT2860AP.dat

9.3.1 AuthMode

Description: WLAN security authentication mode
Value:

AuthMode=OPEN

OPEN	Open system
SHARED	Shared key system
WEPAUTO	Auto switch between OPEN and SHARED
WPAPSK	WPA Pre-Shared Key (Infra)
WPA2PSK	WPA2 Pre-Shared Key (Infra)
WPAPSKWPA2PSK	WPAPSK/WPA2PSK mixed mode (Infra)
WPA	WPA Enterprise mode (Need wpa_supplicant)
WPA2	WPA2 Enterprise mode (Need wpa_supplicant)

WPA1WPA2	WPA/WPA2 mixed mode (Need wpa_supplicant)

9.3.2 EncrypType

Description: WLAN security encryption type
Value:

EncrypType=NONE

NONE	No encryption
WEP	Wired Equivalent Privacy
TKIP	Temporal Key Integrity Protocol
AES	Advanced Encryption Standard
TKIPAES	Mixed cipher

9.3.3 IEEE8021X

Description: Enable or disable 8021X-WEP mode

Value:

IEEE8021X=0

0: disable
1: enable

this field is enabled only when
-WEP or Radius-NONE mode on, otherwise must disable.

9.3.4 RekeyMethod

Description: Configuration of rekey method for WPA/WPA2
Value:

RekeyMethod=DISABLE

TIME:	Time rekey
PKT:	Packet rekey
DISABLE:	Disable rekey

9.3.5 RekeyInterval

Description: Rekey interval configuration for WPA/WPA2
Value:

RekeyInterval=0

The value range is 0 ~ 0x3FFFFF. (Unit: 1 second or 1000 packets)
Use 0 to disable rekey.

9.3.6 PMKCachePeriod

Description: PMK cache life time configuration for WPA/WPA2
Value:

PMKCachePeriod=10

The value range is 0 ~ 65535. (Unit: minute)

9.3.7 WPAPSK

Description: WLAN security password for TKIP/AES
Value:

WPAPSK=01234567

8~63 ASCII characters
64 hexadecimal characters

9.3.8 DefaultKeyID

Description: Default key ID (WEP only)
Value:

DefaultKeyID=1

The ID range is 1~4

9.3.9 Key1Type

Description: Key 1 type
Value:

Key1Type=0

0: Hexadecimal
1: ASCII

9.3.10 Key1Str

Description: Key 1 string
Value:

Key1Str=

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.3.11 Key2Type

Description: Key 2 type
Value:

Key2Type=0

0: Hexadecimal
1: ASCII

9.3.12 Key2Str

Description: Key 2 string

Value:

Key2Str=

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.3.13 Key3Type

Description: Key 3 type

Value:

Key3Type=0

0: Hexadecimal
1: ASCII

9.3.14 Key3Str

Description: Key 3 string

Value:

Key3Str=

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.3.15 Key4Type

Description: Key 4 type

Value:

Key4Type=0

0: Hexadecimal
1: ASCII

9.3.16 Key4Str

Description: Key 4 string

Value:

Key4Str=

10 or 26 hexadecimal characters
5 or 13 ASCII characters

9.3.17 WpaMixPairCipher

Description: Providing more flexible combination of cipher suite
Value:

WpaMixPairCipher=WPA_TKIP_WPA2_AES

WPA_AES_WPA2_TKIPAES
WPA_AES_WPA2_TKIP
WPA_TKIP_WPA2_AES
WPA_TKIP_WPA2_TKIPAES
WPA_TKIPAES_WPA2_AES
WPA_TKIPAES_WPA2_TKIPAES
WPA_TKIPAES_WPA2_TKIP

9.3.18 PreAuth

Description: Enable or disable WPA2 pre-authentication mode
Value:

PreAuth=0

0: disable
1: enable

9.4 New WFA Security Rules

		2013/12/31	2014/1/1
Personal			
WPA-PSK Only	TKIP	V	X
	AES	△	X
WPA2-PSK Only	TKIP	△	X
	AES	V	V
WPA-PSK/WPA2-PSK Mixed			
WPA-PSK	TKIP	V	V
	AES	△	X
WPA2-PSK	TKIP	△	X
	AES	V	V
Enterprise			
WPA Only	TKIP	V	X
	AES	△	X
WPA2 Only	TKIP	△	X
	AES	V	V
WPA/WPA2 Mixed			
WPA	TKIP	V	V
	AES	△	X
WPA2	TKIP	△	X
	AES	V	V

V = Allowed by WFA

X = Prohibited by WFA

△ = It was not prohibited by WFA, but no test case use it.

Note: Please check 9.5.5 for the correct settings of mixed mode.

9.5 iwpriv command examples

Please specify SSID at last step to trigger the AP restart procedure which would reload new security settings.

9.5.1 OPEN/NONE

1. iwpriv ra0 set AuthMode=OPEN
2. iwpriv ra0 set EncrypType=NONE
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=myownssid

9.5.2 SHARED/WEP

1. iwpriv ra0 set AuthMode=SHARED
2. iwpriv ra0 set EncrypType=WEP
3. iwpriv ra0 set Key1=0123456789
4. iwpriv ra0 set DefaultKeyID=1
5. iwpriv ra0 set IEEE8021X=0
6. iwpriv ra0 set SSID=myownssid

9.5.3 WPAPSK/TKIP

1. iwpriv ra0 set AuthMode=WPA2PSK
2. iwpriv ra0 set EncrypType=TKIP
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=myownssid
5. iwpriv ra0 set WPAPSK=myownpresharedkey
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv ra0 set SSID=myownssid

Note: Deprecated by WFA since 2014.01.01

9.5.4 WPA2PSK/AES

1. iwpriv ra0 set AuthMode=WPA2PSK
2. iwpriv ra0 set EncrypType=AES
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set SSID=MySsid
5. iwpriv ra0 set WPAPSK=MyPassword
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv ra0 set SSID=MySsid

9.5.5 WPAPSKWPA2PSK/TKIPAES

1. iwpriv ra0 set AuthMode=WPAPSKWPA2PSK
2. iwpriv ra0 set EncrypType=TKIPAES
3. iwpriv ra0 set IEEE8021X=0
4. iwpriv ra0 set WpaMixPairCipher=**WPA_TKIP_WPA2_AES**
5. iwpriv ra0 set SSID=MySsid
6. iwpriv ra0 set WPAPSK=MyPassword
7. iwpriv ra0 set DefaultKeyID=2
8. iwpriv ra0 set SSID=MySsid

10 Authenticator

rt2860apd - IEEE 802.1X Authenticator (user space utility)
rt2860apd is an optional user space component for the SoftAP driver. It provides IEEE 802.1X Authenticator feature when you choose to use external RADIUS Authentication Server (AS).

10.1 IEEE 802.1X features in rt2860apd

IEEE Std. 802.1X-2001 is a standard for port-based network access control. It introduces an extensible mechanism for authenticating and authorizing users. rt2860apd implements part of IEEE 802.1X features which help AS authorizing Supplicant and also prove itself a valid Authenticator to AS. Please be noted that rt2860apd does not include the state machine for key management. The key management function is included in the SoftAP driver. rt2860apd relays the frames between the Supplicant and the AS. Not until either one timeout or Success or Fail frame indicated does rt2860apd finish the authentication process. The port control entity is implemented in the SoftAP driver.

10.2 How to start rt2860apd

Please run “rt2860apd” in your system script.

10.3 rt2860apd configuration for IEEE 802.1X

When rt2860apd starts, it reads the configuraion file to derive parameters. For any changes to make, one need to first edit the configuration file, then restart rt2860apd.
Please add 4 required parameters in the configuration file for WLAN SoftAP driver (RT2860AP.dat/RT2870AP.dat).
RADIUS_Server='192.168.2.3'
RADIUS_Port='1812'
RADIUS_Key='password'
own_ip_addr='your_ip_addr'
The word in ' ' must be replaced with your own correct setting. Please make sure 'your_ip_addr' and RADIUS_Server is connected and RADIUS_Server's IAS (or related) services are started.

The optional variables as below,

- session_timeout_interval is for 802.1x reauthentication setting.
  - set to zero to disable 802.1x reauthentication service for each session.
  - session_timeout_interval unit is second and must be larger than 60.
  - For example,
    - session_timeout_interval = 120
      reauthenticate each session every 2 minutes.
    - session_timeout_interval = 0
      disable reauthenticate service.
- EAPifname is assigned as the binding interface for EAP negotiation.

- Its default value is "br0". But if the wireless interface doesn't attach to bridge interface or the bridge interface name isn't "br0", please modify it.
- For example,
  - EAPifname=br0

- PreAuthifname is assigned as the binding interface for WPA2 Pre-authentication.
  - Its default value is "br0". But if the ethernet interface doesn't attach to bridge interface or the bridge interface name isn't "br0", please modify it.
  - For example,
    - PreAuthifname=br0

10.4 Support Multiple RADIUS Servers

We use complier option to turn on/off the multiple RADIUS servers for 802.1x.
If you want to enable the feature, make sure that "MULTIPLE_RADIUS" is defined in Makefile. Default is disabled. Besides, you must modify the file "RT2860AP.dat" to co-operate with 802.1x. We extend some variables to support individual RADIUS server IP address, port and secret key for MBSS.
E.g.
RADIUS_Server=192.168.2.1;192.168.2.2;192.168.2.3;192.168.2.4
RADIUS_Port=1811;1812;1813;1814
RADIUS_Key=ralink_1;ralink_2;ralink_3;ralink_4
RADIUS_Server=10.10.10.1; 10.10.10.2; 10.10.10.3; 10.10.10.4
RADIUS_Port=1812;1812;1812;1812
RADIUS_Key=ralink_5;ralink_6;ralink_7;ralink_8

Or
RADIUS_Key1=ralink_1;
RADIUS_Key1=ralink_5;
RADIUS_Key2=ralink_2;
RADIUS_Key2=ralink_6;
RADIUS_Key3=ralink_3;
RADIUS_Key3=ralink_7;
RADIUS_Key4=ralink_4;
RADIUS_Key4=ralink_8;

For backward compatibility, the driver parses "RADIUS_Key" or RADIUS_KeyX"(X=1~4) for radius key usage. But the paramter "RADIUS_Key" has the first priority.
This implies,
The RADIUS server IP of ra0 is 192.168.2.1, its port is 1811 and its secret key is ralink_1.
The RADIUS server IP of ra1 is 192.168.2.2, its port is 1812 and its secret key is ralink_2.
The RADIUS server IP of ra2 is 192.168.2.3, its port is 1813 and its secret key is ralink_3.
The RADIUS server IP of ra3 is 192.168.2.4, its port is 1814 and its secret key is ralink_4. If your wireless interface prefix is not "ra", please modify these variables.

Setup Multiple RADIUS Server failover by iwpriv:

iwpriv ra0 set RADIUS_Server=192.168.1.1;192.168.1.2
iwpriv ra0 set RADIUS_Port=1812;1813
iwpriv ra0 set RADIUS_Key=mediatek123;mediatek456

10.5 Enhance dynamic wep keying

In OPEN-WEP with 802.1x mode, the authentication process generates broadcast and unicast key. The unicast key is unique for every individual client so it is always generated randomly by 802.1x daemon. But the broadcast key is shared for all associated clients; it can be pre-set manually by users or generated randomly by 802.1x daemon.

Through the parameter "DefaultKeyID" and its corresponding parameter "KeyXStr"(i.e. X = the value of DefaultKeyID) in RT2860Ap.dat, the 802.1x daemon would use it as the broadcast key material. But if the corresponding parameter "KeyXStr" is empty or unsuitable, the broadcast key would be generated randomly by the 802.1x daemon.

The 802.1x daemon need to read RT2860AP.dat to decide whether the broadcast key is generated randomly or not, so please update the RT2860AP.dat and restart rt2860apd if those correlative parameters are changed.

10.6 Examples for Radius server configuration

10.6.1 Example I

This is a step-by-step guide to set SoftAP using WPA security mechanism. Assume RT2800 SoftAP has ip address 192.168.1.138, AS (Authentication Server) has IP address 192.168.1.1, Radius Secret is myownkey.

1. load WLAN SoftAP driver
   - $insmod rt2860ap.o
2. First edit configuration file with correct value, esp. the following parameters that relate to the authentication features of RT2800AP.dat
   RADIUS_Server=192.168.1.1
   RADIUS_Port=1812
   RADIUS_Key=myownkey
   own_ip_addr=192.168.1.138
3. start RT2800apd daemon by typing.
   - $rt2860apd
4. iwpriv ra0 set AuthMode=WPA
5. iwpriv ra0 set EncrypType=TKIP
6. iwpriv ra0 set DefaultKeyID=2
7. iwpriv ra0 set IEEE8021X=0
8. iwpriv ra0 set SSID=myownssid

10.6.2 Example II

Change 802.1x settings to WPA with TKIP, using 802.1x authentication.

1. Modify 4 parameters
   RADIUS_Server=192.168.2.3
   RADIUS_Port=1812
   RADIUS_Key=password
   own_ip_addr=192.168.1.123

in the RT2860AP.dat and save.

2. iwpriv ra0 set AuthMode=WPA
3. iwpriv ra0 set EncrypType=TKIP
4. iwpriv ra0 set IEEE8021X=0
5. iwpriv ra0 set SSID=myownssid

Note:

Step 4 restarts the rt2860apd, and is essential.

10.6.3 Example III

Change setting to OPEN/WEP with 802.1x.

1. iwpriv ra0 set AuthMode= OPEN
2. iwpriv ra0 set EncrypType= WEP
3. iwpriv ra0 set IEEE8021X=1

Note:

"IEEE8021X=1" only when Radius-WEP or Radius-NONE mode on, otherwise must "IEEE8021X=0".

10.6.4 Example V

Change setting to OPEN/NONE with 802.1x.

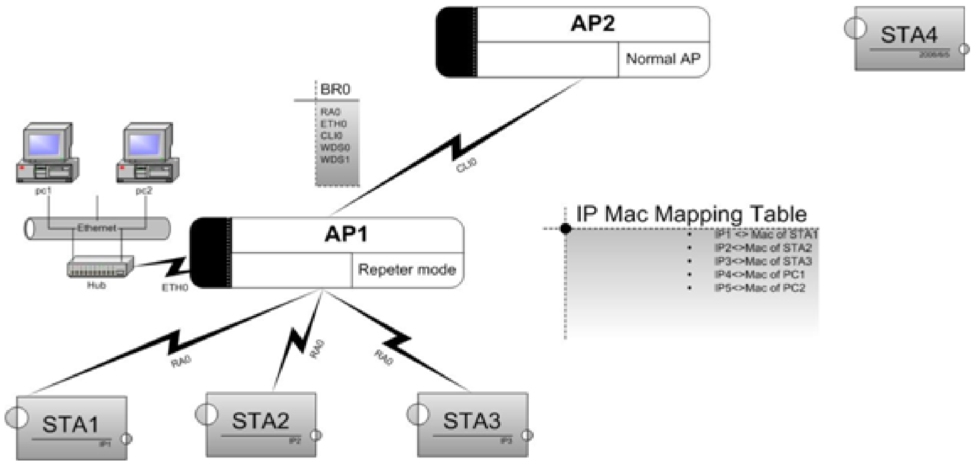
1. iwpriv ra0 set AuthMode= OPEN
2. iwpriv ra0 set EncrypType= NONE
3. iwpriv ra0 set IEEE8021X=1

Note:

"IEEE8021X=1" only when Radius-WEP or Radius-NONE mode on , otherwise must "IEEE8021X=0".

11 AP-CLIENT

The AP-Client function provides a simulated and virtual STA interface while the original AP interface is working simultaneously. Its application is usually a wireless repeater or a wireless extender. AP-Client mainly provides a 1-to-N MAC address mapping mechanism such that multiple stations connected to the AP can transparently communicate with another AP, which we usually call RootAP. When AP-Client function is enabled, besides the original AP interface named ra0, a virtual interface named apcli0 will be created. In a repeater application, the software bridge, like br0, is used to relay packets between these two interfaces.The following figure shows the common network topology and operation module of our AP-Client function.



AP1 is an Access Point which enabled AP-Client and therefore has two wireless interfaces, ra0 and apcli0, providing the AP and station function respectively. AP2 is just a tranditional Access Point that provides normal AP function. In the figure, you can see that STA1 associated to AP1 and STA4 associated to AP2. In the old days, if STA1 wants to communicate with STA4, AP1 and AP2 must have some kind of connection between them to relay traffic, like Ethernet LAN (wired) or WDS (wireless). Now with the new AP-Client feature, AP1 can use the simulated STA interface apcli0 to connect to AP2, thus creating the link, and then STA1 can communicate with STA4 transparently and wired stations connected to AP1 through Ethernet could also communicate with STA4.

Here are some reminders for you before using AP-Client.

- AP-Client only supports the following protocols due to the limitation of 1-to-N MAC address mapping mechanism
  - All IP-based network applications
  - ARP
  - DHCP
  - PPPoE
- The last hexadecimal number of the MAC address must be a multiple of 2

11.1 How to Setup AP-Client

- Turn on **APCLI_SUPPORT** in driver config
- Use “**ifconfig apcli0 up**” to bring up your AP-Client interface
- In a repeater application, you may use the following commands to bridge ra0 and apcli0

  - **brctl addif br0 ra0**
  - **brctl addif br0 apcli0**
- The security policy support for AP-Client include
  - OPEN
  - SHARED (WEP)
  - WPAPSK (TKIP, AES)
  - WPA2PSK (TKIP, AES)
- Please be noted that AP-Client is also a virtual interface. When you use AP-Client with MBSSID simultaneously, AP-Client will consume one position and the parameter “BssidNum” should be larger than 1 and less than 7 (1 < BssidNum < 7)
- Use “**iwpriv apcli0 show connStatus**” to display connection status with RootAP

11.2 Parameters in RT2860AP.dat

11.2.1 ApCliEnable

Description: Enable or disable AP-Client function
Value:

ApCliEnable=1

0: disable
1: enable

11.2.2 ApCliSsid

Description: Configure the target/RootAP SSID which AP-Client wants to connect with
Value:

ApCliSsid=target_ssid

target_ssid: 1~32 characters

11.2.3 ApCliBssid

Description: Configure the target BSSID which AP-Client wants to join
Value:

ApCliBssid=00:11:22:33:44:55

Note: It is an optional command. Users can use this command to indicate the desired BSSID. Otherwise, AP-Client would get correct BSSID according to configured SSID automatically.

11.2.4 ApCliAuthMode

Description: AP-Client authentication mode configuration
Value:

ApCliAuthMode=OPEN

OPEN
SHARED
WPAPSK
WPA2PSK

11.2.5 ApCliEncrypType

Description: AP-Client encryption type configuration
Value:

ApCliEncrypType=NONE

NONE
WEP
TKIP
AES

11.2.6 ApCliWPAPSK

Description: WPA/WPA2 Pre-Shared Key configuration
Value:

ApCliWPAPSK=12345678

8~63 ASCII characters
64 hexadecimal characters

11.2.7 ApCliDefaultKeyID

Description: Default key index configuration
Value:

ApCliDefaultKeyID=1

The ID range is 1~4

11.2.8 ApCliKey1Type

Descripion: Set the WEP key type of AP-Client for key index 1
Value:

ApCliKey1Type=0

0: Hexadecimal
1: ASCII

11.2.9 ApCliKey1Str

Description: Set the WEP key string of AP-Client for key 1
Value:

ApcliKey1Str=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.2.10 ApCliKey2Type

Descripion: Set the WEP key type of AP-Client for key index 2
Value:

ApCliKey2Type=0

0: Hexadecimal
1: ASCII

11.2.11 ApCliKey2Str

Description: Set the WEP key string of AP-Client for key 2
Value:

ApcliKey2Str=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.2.12 ApCliKey3Type

Description: Set the WEP key type of AP-Client for key index 3
Value:

ApCliKey3Type=0

0: Hexadecimal
1: ASCII

11.2.13 ApCliKey3Str

Description: Set the WEP key string of AP-Client for key 3
Value:

ApcliKey3Str=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.2.14 ApCliKey4Type

Description: Set the WEP key type of AP-Client for key index 4
Value:

ApCliKey4Type=0

0: Hexadecimal
1: ASCII

11.2.15 ApCliKey4Str

Description: Set the WEP key string of AP-Client for key 4
Value:

ApcliKey4Str=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.2.16 ApCliTxMode

Description: Fixed transmission mode configuration
Value:

ApCliTxMode=HT

cck|CCK,
ofdm|OFDM,
ht|HT

11.2.17 ApCliTxMcs

Description: AP-Client Tx MCS configuration
Value:

ApCliTxMcs=33

0~15, 32: Fixed MCS
33: Auto MCS

11.2.18 ApCliWscSsid

Description: Configure the SSID which AP-Client wants to do WPS negotiation
Value:

ApCliWscSsid=target_ssid

targer_ssid: 1~32 characters

11.3 AP-Client iwpriv command

11.3.1 ApCliEnable

Description: Enable or disable AP-Client function
Value:

iwpriv apcli0 set ApCliEnable=0

0: disable
1: enable

11.3.2 ApCliSsid

Description: Configure the target/RootAP SSID which AP-Client wants to connect with
Value:

iwpriv apcli0 set ApCliSsid=target_ssid

target_ssid: 1~32 characters

11.3.3 ApCliBssid

Description: Configure the target BSSID which AP-Client wants to join
Value:

iwpriv apcli0 set ApCliBssid=00:11:22:33:44:55

Note: It is an optional command. Users can use this command to indicate the desired BSSID. Otherwise, AP-Client would get correct BSSID according to configured SSID automatically.

11.3.4 ApCliAuthMode

Description: AP-Client authentication mode configuration
Value:

iwpriv apcli0 set ApCliAuthMode=OPEN

OPEN
SHARED
WPAPSK
WPA2PSK

11.3.5 ApCliEncrypType

Description: AP-Client encryption type configuration
Value:

iwpriv apcli0 set ApCliEncrypType=NONE

NONE
WEP
TKIP
AES

11.3.6 ApCliWPAPSK

Description: WPA/WPA2 Pre-Shared Key configuration
Value:

iwpriv apcli0 set ApCliWPAPSK=12345678

8~63 ASCII characters
64 hexadecimal characters

11.3.7 ApCliDefaultKeyID

Description: Default key index configuration
Value:

iwpriv apcli0 set ApCliDefaultKeyID=1

The ID range is 1~4

11.3.8 ApCliKey1

Description: Set the WEP key string of AP-Client for key 1
Value:

iwpriv apcli0 set ApcliKey1=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.3.9 ApCliKey2

Description: Set the WEP key string of AP-Client for key 2
Value:

iwpriv apcli0 set ApcliKey2=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.3.10 ApCliKey3

Description: Set the WEP key string of AP-Client for key 3
Value:

iwpriv apcli0 set ApcliKey3=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.3.11 ApCliKey4

Description: Set the WEP key string of AP-Client for key 4
Value:

iwpriv apcli0 set ApcliKey4=012345678

10 or 26 hexadecimal characters
5 or 13 ASCII characters

11.3.12 ApCliTxMode

Description: Fixed transmission mode configuration
Value:

iwpriv apcli0 set ApCliTxMode=HT

CCK
OFDM
HT

11.3.13 ApCliTxMcs

Description: AP-Client Tx MCS configuration
Value:

iwpriv apcli0 set ApCliTxMcs=33

0~15, 32:	Fixed MCS
33:	Auto MCS

11.3.14 ApCliWscSsid

Description: Configure the SSID which AP-Client wants to do WPS negotiation
Value:

iwpriv apcli0 set ApCliWscSsid=target_ssid

targer_ssid: 1~32 characters

11.3.15 ApCliAutoConnect

Description: Enable or disable the auto-connection function to find the configured SSID
Value:

iwpriv ra0 set ApCliAutoConnect=1

0: disable
1: enable

Note: APCLI_AUTO_CONNECT_SUPPORT must be turned on

11.4 AP-Client normal connection examples

11.4.1 OPEN/NONE

iwpriv apcli0 set ApCliEnable=0
iwpriv apcli0 set ApCliAuthMode=OPEN
iwpriv apcli0 set ApCliEncrypType=NONE
iwpriv apcli0 set ApCliSsid=ROOTAP_SSID
iwpriv apcli0 set ApCliEnable=1

11.4.2 OPEN/WEP

iwpriv apcli0 set ApCliEnable=0
iwpriv apcli0 set ApCliAuthMode=OPEN
iwpriv apcli0 set ApCliEncrypType=WEP
iwpriv apcli0 set ApCliDefaultKeyID=1
iwpriv apcli0 set ApCliKey1=1234567890
iwpriv apcli0 set ApCliSsid=ROOTAP_SSID
iwpriv apcli0 set ApCliEnable=1

11.4.3 WPAPSK/TKIP

iwpriv apcli0 set ApCliEnable=0
iwpriv apcli0 set ApCliAuthMode=WPAPSK
iwpriv apcli0 set ApCliEncrypType=TKIP
iwpriv apcli0 set ApCliSsid=ROOTAP_SSID
iwpriv apcli0 set ApCliWPAPSK=12345678
iwpriv apcli0 set ApCliSsid=ROOTAP_SSID
iwpriv apcli0 set ApCliEnable=1

11.4.4 WPA2PSK/AES

iwpriv apcli0 set ApCliEnable=0
iwpriv apcli0 set ApCliAuthMode=WPA2PSK
iwpriv apcli0 set ApCliEncrypType=AES
iwpriv apcli0 set ApCliSsid=ROOTAP_SSID
iwpriv apcli0 set ApCliWPAPSK=12345678
iwpriv apcli0 set ApCliSsid=ROOTAP_SSID
iwpriv apcli0 set ApCliEnable=1

11.5 AP-Client WPS connection examples

11.5.1 PIN mode

iwpriv apcli0 set ApCliEnable=1
iwpriv apcli0 set WscConfMode=1 // Enrollee

```
iwpriv apcli0 set WscMode=1          // PIN mode
iwpriv apcli0 set WscGetConf=1       // Trigger
iwpriv apcli0 set ApCliEnable=1
```

11.5.2 PBC Mode

```
iwpriv apcli0 set ApCliEnable=1
iwpriv apcli0 set WscConfMode=1      // Enrollee
iwpriv apcli0 set WscMode=2          // PBC mode
iwpriv apcli0 set WscGetConf=1       // Trigger
iwpriv apcli0 set ApCliEnable=1
```

12 WDS

A Wireless Distribution System is a system enabling the wireless interconnection of acess points. Each WDS AP needs to be in the same channel, using the same encryption type.

Actually, there is no test plan to ensure the inter-operability of all WDS products from different Vendors. Mediatek's implementation provides two modes of AP-to-AP connectivity. One is **Bridge mode**, in which WDS APs communicate only with each other and do not allow wireless stations to access them. The other is **Repeater mode**, in which WDS APs communicate with each other and with wireless stations.

In case you want to have an auto-learning WDS peer, we also provide the **Lazy mode** in which you do not need to thoroughly configure the WDS settings. However, be noted that you cannot configure all APs to be in Lazy mode, otherwise no 4-address frame will be transmitted at all and auto-learning would be impossible. This means that there should be at least one AP being configured to Bridge mode or Repeater mode.

12.1 How to Steup WDS

1. Edit the driver profile in each WDS peer

   WDS Peer-A with the MAC address 00:0C:43:aa:bb:cc
   - WdsEnable=3
   - WdsPhyMode=HTMIX;
   - WdsList=00:0C:43:11:22:33;
   - WdsEncrypType=NONE;

   WDS Peer-B with the MAC address 00:0C:43:11:22:33
   - WdsEnable=3
   - WdsPhyMode=HTMIX;
   - WdsList=00:0C:43:aa:bb:cc;
   - WdsEncrypType=NONE;

2. Edit your networking script file, like bridge_setup.sh, according to the number of WDS link. Add "**brctl addif br0 wds0**" and "**ifconfig wds0 0.0.0.0**" to relative places

3. Use "**iwpriv ra0 show wdsinfo**" to display WDS link information

12.2 WDS Security

WDS security is **PSK-only**, and it does not support mixed mode, like WPAPSKWPA2PSK.

When WDS is in Lazy mode, all WDS links (wds0 ~ wds3) shall share the same encryption type and key material (referring to wds0 settings). Otherwise, each WDS link has its own security settings. No matter what WDS mode you use, it has nothing to do with the encryption of the main BSSID (ra0).

WdsKey:
It is used for all WDS interfaces and supports only AES and TKIP configuration. If you want to use WEP, key settings will be retrieved from the main BSSID.

Wds0Key/Wds1Key/Wds2Key/Wds3Key:
They are used to configure key settings for each WDS interface.

The following example is to create one WDS link (wds0) with AES encryption.
WdsEnable=3
WdsPhyMode=HTMIX;HTMIX;HTMIX;HTMIX
WdsList=00:0c:43:12:34:56;
WdsEncrypType=AES;NONE;NONE;NONE
Wds0Key=12345678
Wds1Key=
Wds2Key=
Wds3Key=

12.3 Parameters in RT2860AP.dat

12.3.1 WdsEnable

Description: WDS function configuration
Value:

WdsEnable=0

0: **Disable** - Disable WDS function.
1: Restrict mode - Same as Repeater mode.
2: **Bridge mode** - Enable WDS and work like a bridge.
The MAC address of peer WDS APs should be configured in the "WdsList" field.
In this mode, AP is just a bridge and will not send any beacon and will not respond to any probe request packet. Therefore STA will not be able to connect with it.
3: **Repeater mode** - Enable WDS and work like a repeater.
The MAC address of peer WDS APs should be configured in the "WdsList" field.
4: **Lazy mode** - Enable WDS function.
It automatically learns from 4-address format frames sent by the WDS peer and you do not have to configure WdsList manually.

12.3.2 WdsList

Description: WDS peer MAC address configuration
Value:

WdsList=00:10:20:30:40:50;

The maximum WDS link number is 4.
wds0;wds1;wds2;wds3

12.3.3 WdsEncrypType

Description: WDS encryption configuration
Value:

WdsEncrypType=NONE;

The option includes NONE, WEP, TKIP and AES.

Example:

WdsEncrypType=OPEN;WEP;TKIP;AES

The encrptytion of wds0 is OPEN
The encrptytion of wds1 is WEP
The encrptytion of wds2 is TKIP
The encrptytion of wds3 is AES

12.3.4 WdsKey

Description: WDS key configuration

Value:

WdsKey=12345678

8 ~ 63 ASCII characters (eg: 12345678) for TKIP or AES
64 hexadecimal characters for TKIP or AES

WdsKey is kept for backward-compatibility and it only supports TKIP and AES.
You can use either WdsKey or Wds[0-4]Key but not both.

Note: Combinations of WDS security mode

EncrypType	WdsEncrypType	WdsEncrypType of the WDS peer	Note
NONE	NONE	NONE	
WEP	WEP	WEP	Using legacy key setting method
TKIP	TKIP	TKIP	WDS's key is from WdsKey
TKIP	AES	AES	WDS's key is from WdsKey
AES	TKIP	TKIP	WDS's key is from WdsKey
AES	AES	AES	WDS's key is from WdsKey
TKIPAES	TKIP	TKIP	WDS's key is from WdsKey
TKIPAES	AES	AES	WDS's key is from WdsKey

12.3.5 Wds0Key

Description: WDS key for Link-0

Value:

Wds0Key=12345678

10 or 26 hexadecimal characters (eg: 1234567890) for WEP
5 or 13 ASCII characters (eg: 12345) for WEP
8 ~ 63 ASCII characters (eg: 12345678) for TKIP or AES
64 hexadecimal characters for TKIP or AES

12.3.6 Wds1Key

Description: WDS key for Link-1

Value:

Wds1Key=12345678

10 or 26 hexadecimal characters (eg: 1234567890) for WEP
5 or 13 ASCII characters (eg: 12345) for WEP
8 ~ 63 ASCII characters (eg: 12345678) for TKIP or AES
64 hexadecimal characters for TKIP or AES

12.3.7 Wds2Key

Description: WDS key for Link-2
Value:

Wds2Key=12345678

10 or 26 hexadecimal characters (eg: 1234567890) for WEP
5 or 13 ASCII characters (eg: 12345) for WEP
8 ~ 63 ASCII characters (eg: 12345678) for TKIP or AES
64 hexadecimal characters for TKIP or AES

12.3.8 Wds3Key

Description: WDS key for Link-3
Value:

Wds3Key=12345678

10 or 26 hexadecimal characters (eg: 1234567890) for WEP
5 or 13 ASCII characters (eg: 12345) for WEP
8 ~ 63 ASCII characters (eg: 12345678) for TKIP or AES
64 hexadecimal characters for TKIP or AES

12.3.9 WdsPhyMode

Description: WDS link physical mode configuration
Value:

WdsPhyMode=HTMIX;

The option includes CCK, OFDM, HTMIX and GREENFIELD.

Example:

WdsPhyMode=CCK;OFDM;HTMIX;GREENFIELD

The PHY mode of wds0 is CCK
The PHY mode of wds1 is OFDM
The PHY mode of wds2 is HTMIX
The PHY mode of wds3 is GREENFIELD

13 IGMP SNOOPING

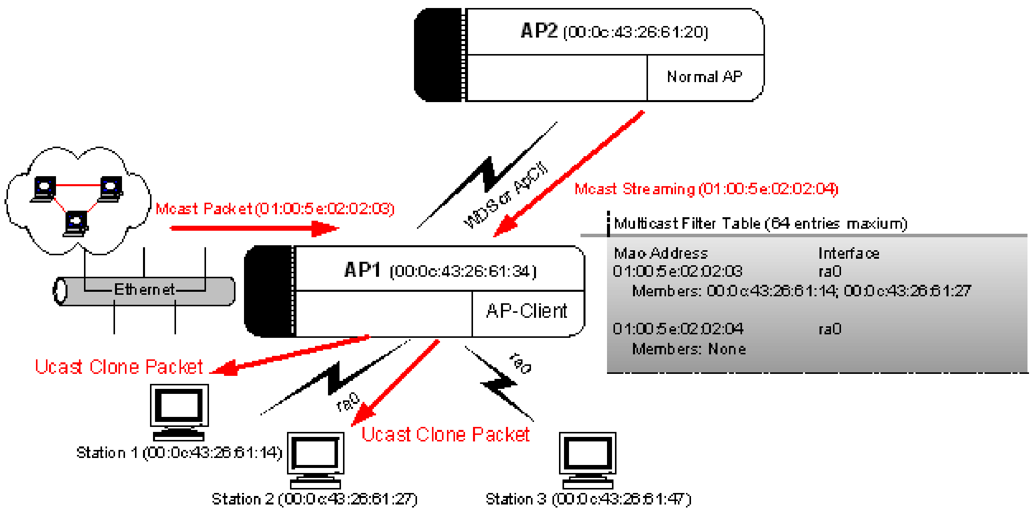
13.1 Basic

Please check the following two Wiki links.

- http://en.wikipedia.org/wiki/Multicast_address
- http://en.wikipedia.org/wiki/Internet_Group_Management_Protocol

IGMP Snooping provides a mechanism converting multicast traffic into unicast traffic. When AP receives incoming multicast traffic, the conversion would be done based on an IGMP Snooping table (Multicast Filter Table).

13.2 Introduction to IGMP Snooping Table



An IGMP Snooping table (a.k.a. Multicast Filter Table) entry consists of 3 components, Group-ID (Multicast MAC Address), Network-Interface and Member-List. Taking above figure for example, you can see that Multicast Filter Table of AP1 has two table entries. One is “01:00:5e:02:02:03” with two members on interface ra0 and the other is “01:00:5e:02:02:04” without any member on interface ra0.

In our implementation, AP will automatically maintain the Multicast Filter Table through packet snooping. The IGMP-Membership-Report packets sent from connected stations would be checked and parsed. You can also manually add and delete an entry through iwpriv command.

13.3 Multicast Packet Parsing Process

When AP receives multicast packets, it will check whether the multicast destination address matches any Group-ID in the Multicast Filter Table. AP will drop the packet if no match found. Otherwise, there are two cases how AP handles a multicast packet. The first one is that Member-List of the matching entry is empty and then AP just forwards multicast packets to all stations connected to the Network-

Interface. In the second case, there are members in the Member-List and AP will do the MC-to-UC conversion based on the membership.

Taking the previous figure for example, AP1 received an Ethernet multicast packet with Group-ID being 01:00:5e:02:02:03. Firstly AP1 checked the Multicast Filter Table and found the first entry matched. Therefore, AP1 cloned every multicast packet into two unicast packets destined to Station 1 and Station 2 respectively.

In the same figure, a multicast streaming sent from AP2 to AP1 with Group-ID 01:00:5e:02:02:04 was forwarded to all stations connected to AP1 (ra0) since the matching entry had no member at all.

<Multicast Filter Table Example>

Group-ID	Network-Interface	Member-List
01:00:5e:02:02:03	ra0	00:0c:43:26:61:14 (Station 1) 00:0c:43:26:61:27 (Station 2)
01:00:5e:02:02:04	ra0	

13.4 Parameters in RT2860AP.dat

13.4.1 IgmpSnEnable

Description: Enable or disable IGMP Snooping function
Value:

IgmpSnEnable=1

0: disable
1: enable

Note: Please make sure that IGMP_SNOOP_SUPPORT is turned on in driver config

13.5 IGMP Snooping iwpriv command

13.5.1 IgmpSnEnable

Description: Enable or disable IGMP Snooping function
Value:

iwpriv ra0 set IgmpSnEnable=1

0: disable
1: enable

13.5.2 IgmpAdd

Description: Create a new group or add a new member to the existing group
Format:

// Create a new group <Group-ID> which can be a MAC address or an IP address

iwpriv ra0 set IgmpAdd=<Group-ID>
// Add a new member to the existing group. [Member] can only be a MAC address
iwpriv ra0 set IgmpAdd=<Group-ID-[Member]-…>

Value:

// Create a new group via either IP or MAC address
iwpriv ra0 set IgmpAdd=**226.2.2.3**
iwpriv ra0 set IgmpAdd=**01:00:5e:02:02:03**

// Add a new member to the existing group
iwpriv ra0 set IgmpAdd=**226.2.2.3-00:0c:43:26:61:11**
// Add 2 new members to the existing group
iwpriv ra0 set IgmpAdd=**01:00:5e:02:02:03-00:0c:43:26:61:27-00:0c:43:26:61:28**

13.5.3 IgmpDel

Description: Delete a group or remove a member from the existing group
Format:

// Delete a group <Group-ID> which can be a MAC address or an IP address
iwpriv ra0 set IgmpDel=<Group-ID>
// Remove a member from the existing group. [Member] can only be a MAC address
iwpriv ra0 set IgmpDel=<Group-ID-[Member]-…>

Value:

// Delete a new group via either IP or MAC address
iwpriv ra0 set IgmpDel=**226.2.2.3**
iwpriv ra0 set IgmpDel=**01:00:5e:02:02:03**

// Remove a member from the existing group
iwpriv ra0 set IgmpDel=**226.2.2.3-00:0c:43:26:61:11**

// Remove members from the existing group
iwpriv ra0 set IgmpDel=**01:00:5e:02:02:03-00:0c:43:26:61:27-00:0c:43:26:61:28**

14 MAC Repeater

The MAC Repeater is a variation of the original AP-Client funcation and it acts as a wireless proxy for its clients. The repeater will create a corresponding upstream connection to the RootAP for each downstream client connected to it. An upstream connection is created according to its own wireless capability and security mode. When a client disconnects from the repeater, the repeater must also disconnect its corresponding upstream connection with the RootAP. All communication between downstream clients and upstream RootAP utilizes one "AP-Client" interface on the repeater.

For example, if there are 3 clients connected to the repeater, 3 upstream connections will be created accordingly. Besides these "proxy connection", the repeater itself would also create a connection with RootAP. Therefore, in this case there would be totally 3 downstream and 4 upstream connections.

Please be noted that MAC Repeater has the following limitation.

- Roaming of STAs between different BSSs is not supported
- WPA2-Enterprise Security is not supported
- Supported protocols: IPv4 / ARP / DHCP
- The MAC Repeater supports up to 16 clients
- Impact CPU utilization due to parsing all received packets from the STA and all multicast and broadcast packets

14.1 MAC Repeater iwpriv command

14.1.1 MACRepeaterEn

Description: Enable or disable MAC Repeater function
Value:

iwpriv ra0 set MACRepeaterEn=1

0: disable
1: enable

14.1.2 Example

- **iwpriv ra0 set MACRepeaterEn=1**
- ifconfig apcli0 up
- brctl addif br0 apcli0
- iwpriv apcli0 set ApCliEnable=0
- iwpriv apcli0 set ApCliAuthMode=OPEN
- iwpriv apcli0 set ApCliEncrypType=NONE
- iwpriv apcli0 set ApCliSsid=RootAP_SSID
- iwpriv apcli0 set ApCliEnable=1

14.2 Parameter in RT2860AP.dat

14.2.1 MACRepeaterEn

Description: Enable or disable the MAC Repeater function.

Value:

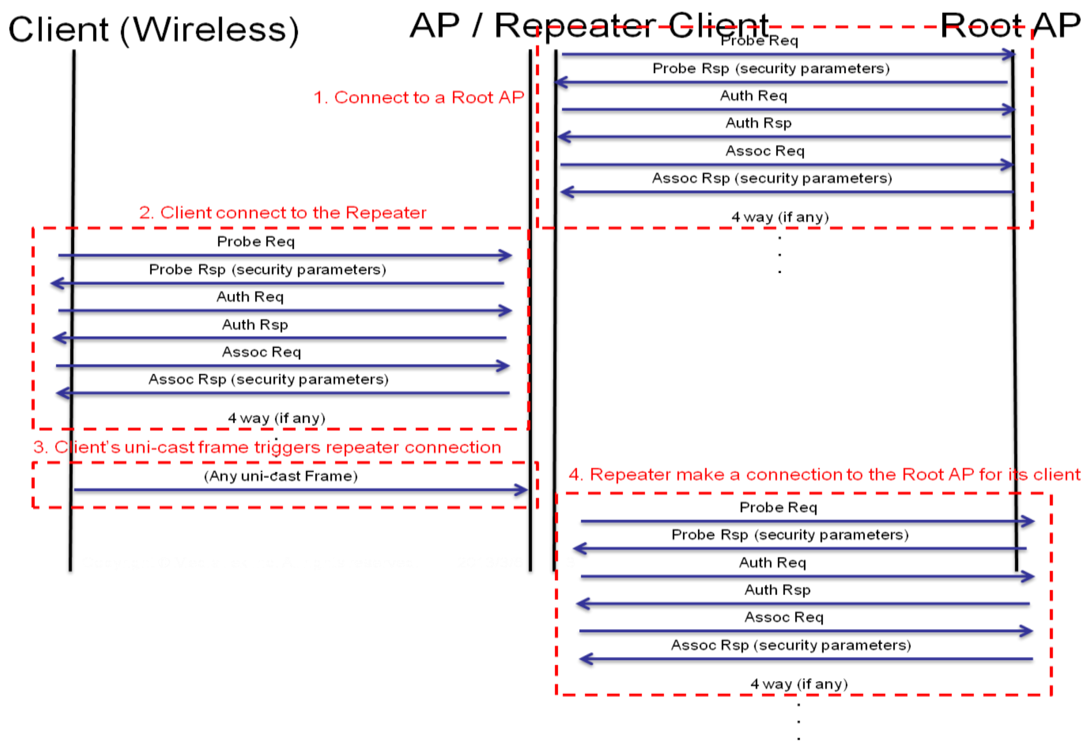
MACRepeaterEn=0

0: disable

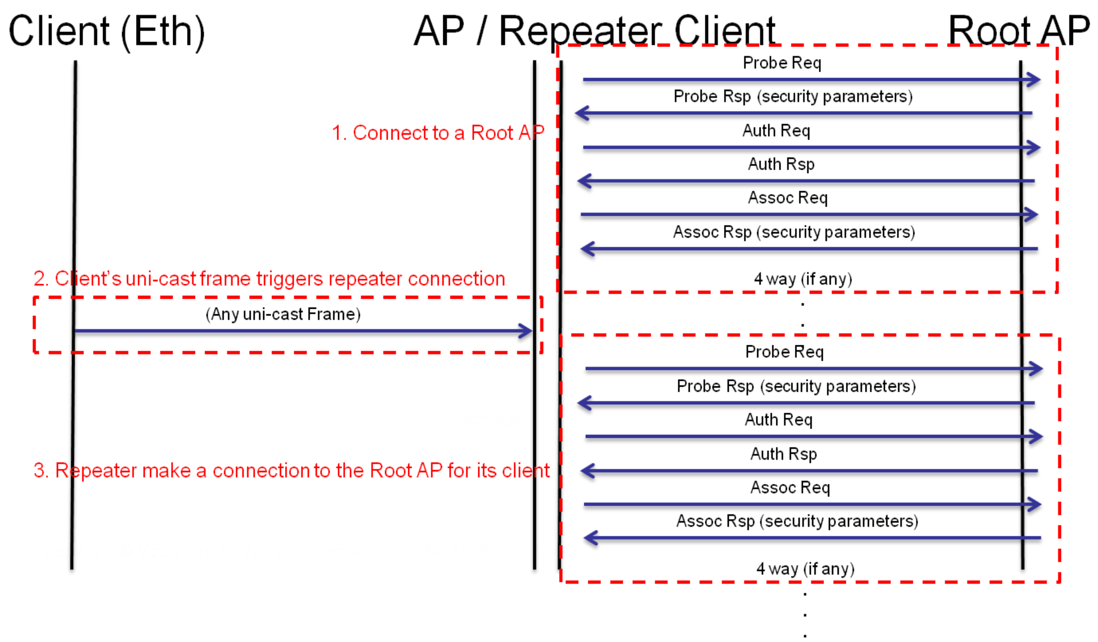
1: enable

14.3 Management Frame Flow

14.3.1 Wireless client

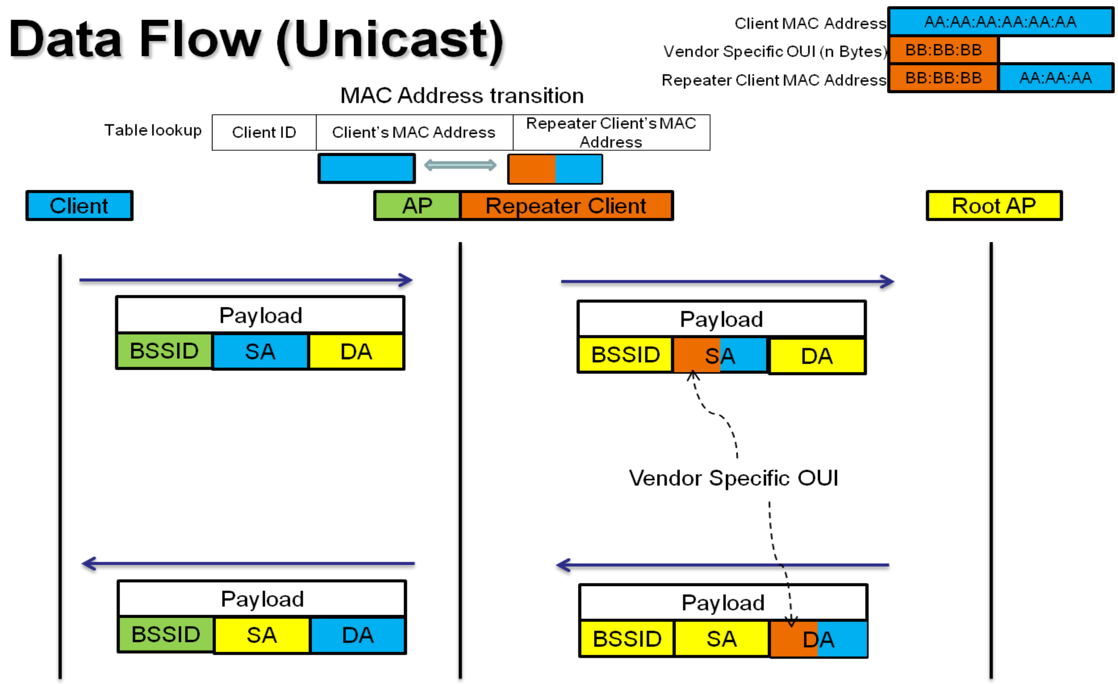


14.3.2 Ethernet client

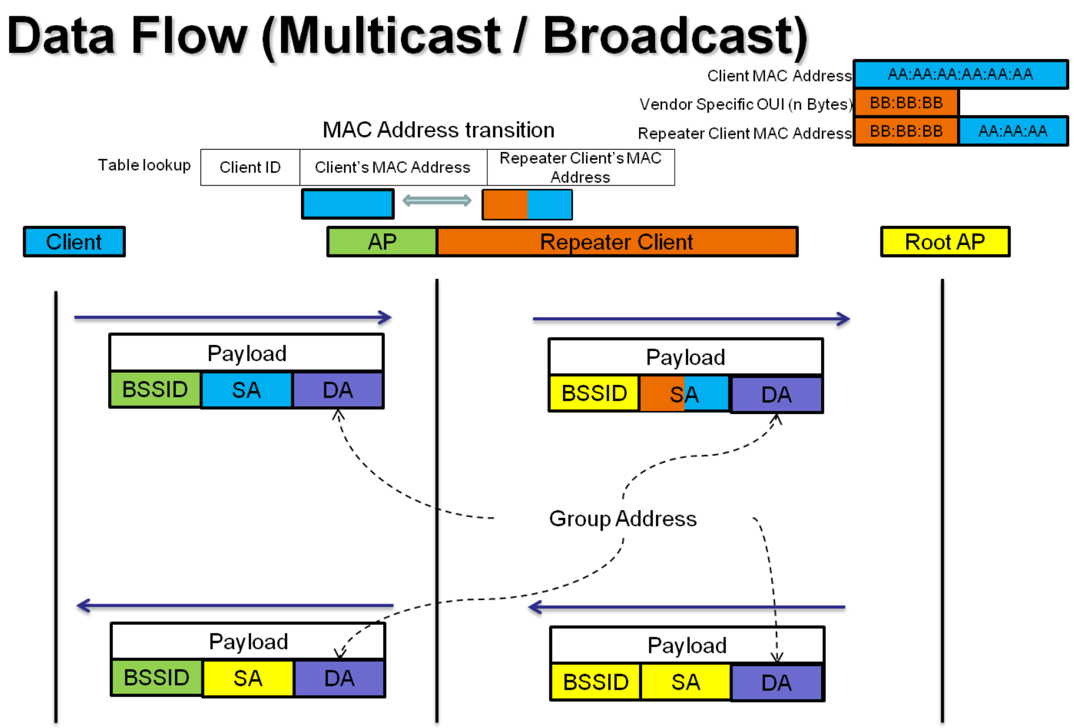


14.4 Data Frame Flow

14.4.1 Unicast



14.4.2 Multicast / Broadcast



15 PMF

PMF stands for Protected Management Frame and IEEE 802.11w is the PMF standard. Its objective is to increase the security of 802.11 management frames.

15.1 PMF iwpriv command

15.1.1 PMFMFPC

Description: Enable or disable Protection Management Frame Capable
Value:

iwpriv ra0 set PMFMFPC=1

0: disable
1: enable

15.1.2 PMFMFPR

Description: Enable or disable Protection Management Frame Required
Value:

iwpriv ra0 set PMFMFPR=1

0: disable
1: enable

15.1.3 PMFSHA256

Description: Enable or disable use SHA256 for Encryption
Value:

iwpriv ra0 set PMFSHA256=1

0: disable
1: enable

Note: SHA stands for Secure Hash Algorithm

15.2 Parameters in RT2860AP.dat

15.2.1 PMFMFPC

Description: Disable or enable Protection Management Frame Capable
Value:

PMFMFPC=0

0: Disable
1: Enable

15.2.2 PMFMFPR

Description: Disable or enable Protection Management Frame Required
Value:

PMFMFPR=0

0: Disable
1: Enable

15.2.3 PMFSHA256

Description: Disable or enable use SHA256 for Encryption
Value:

PMFSHA256=0

0: Disable
1: Enable

15.3 Wi-Fi PMF Testing Note

15.3.1 DUT Requirement

PMF is a mandatory testing item to TGac but an optional one to TGn. Actually you can refer to the following table for the correct combination in a dual band router.

Combination	11ac 5GHz	11n 5GHz	11n 2.4GHz
Correct	PMF supported	PMF supported	PMF supported
Not acceptable	PMF supported	PMF supported	PMF Not Available
Correct	PMF supported	PMF Not Available	PMF Not Available
Not acceptable	PMF supported	PMF Not Available	PMF supported

15.3.2 PMF Test Section 4.3.3.3

Verification of CCMP to protect transmitted unicast deauthentication/disassociation frames

- iwpriv ra0 set PMFMFPC=1
- iwpriv ra0 set PMFMFPR=0
- iwpriv ra0 set PMFSHA256=0
- iwpriv ra0 set SSID=PMF-4.3.3.3
- iwpriv ra0 set **DisConnectSta=00:0C:43:35:93:00**

15.3.3 PMF Test Section 4.4

Verify use of BIP (Broadcast Integrity Protocol) to protect broadcast management frames

- iwpriv ra0 set PMFMFPC=1
- iwpriv ra0 set PMFMFPR=0
- iwpriv ra0 set PMFSHA256=0
- iwpriv ra0 set SSID=PMF-4.4
- iwpriv ra0 set **DisConnectAllSta=2**

16 MBSSID

The Multiple BSSID (MBSSID) function is a feature providing additional virtual WLANs which look like real WLANs to users. Its common application is to create one Main and several Guest Networks simultaneously. You may setup each BSSID with different configuration.

16.1 How to Setup

Please turn on MBSS_SUPPORT in driver config.

```
[*]     MBSSID
[*]       New MBSSID MODE
```

We also suggest turn on NEW_MBSSID_MODE which changes how the driver creates extended MAC addresses for these virtual BSSID.

16.2 Parameter in RT2860AP.dat

16.2.1 BssidNum

Description: Multiple BSSID number configuration
Value:

BssidNum=1

1/2/4/8/16

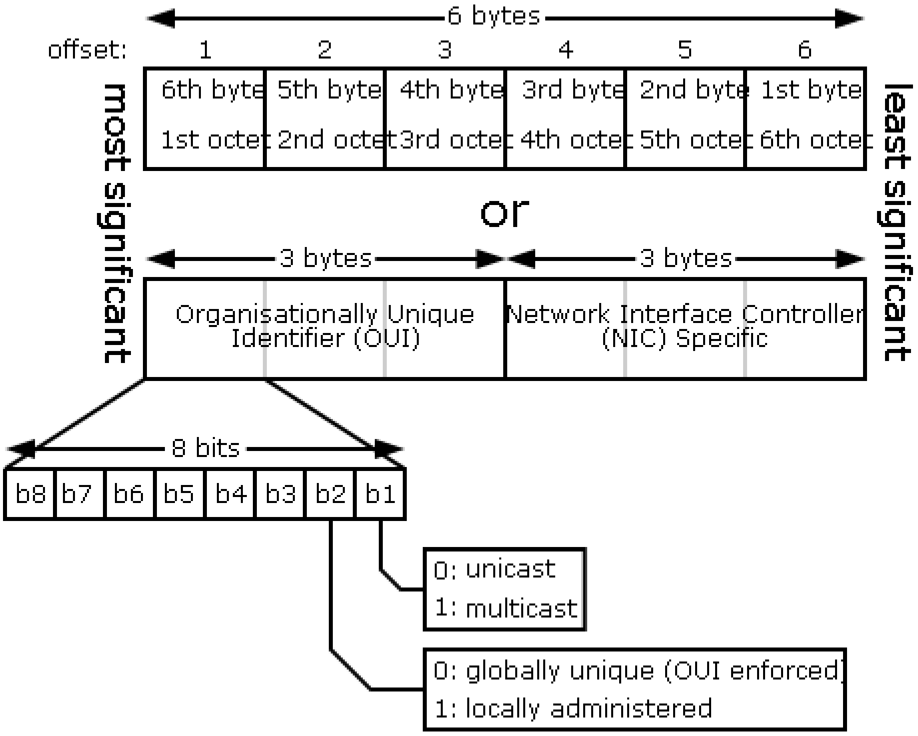
Note:

1. It depends on MBSS_SUPPORT
2. It should be placed before other configuration in the profile
3. 16-BSSID is supported only in new products

16.3 Important Note

16.3.1 MAC Address Format

The following MAC address format figure is from http://en.wikipedia.org/wiki/MAC_address and all subsequent discussion is based on this format.



16.3.2 Old MBSSID Mode

As to main BSSID, **the 1st byte** of its MAC address should be:

- Multiple of 2 for 2-BSSID
- Multiple of 4 for 4-BSSID
- Multiple of 8 for 8-BSSID

Taking BssidNum=4 for example, address extension would be done on 1st byte.

- ra0: 00:0c:43:00:00:0**0** 00 is multiple of 4
- ra1: 00:0c:43:00:00:0**1** 01 comes from (1st byte 0x00) + 1
- ra2: 00:0c:43:00:00:0**2** 02 comes from (1st byte 0x00) + 2
- ra3: 00:0c:43:00:00:0**3** 03 comes from (1st byte 0x00) + 3

Other possible address extension:

Multiple of 4	1st BSSID	2nd BSSID	3rd BSSID	4th BSSID
0x00	AA-BB-CC-DD-EE-F0	AA-BB-CC-DD-EE-F1	AA-BB-CC-DD-EE-F2	AA-BB-CC-DD-EE-F3
0x04	AA-BB-CC-DD-EE-F4	AA-BB-CC-DD-EE-F5	AA-BB-CC-DD-EE-F6	AA-BB-CC-DD-EE-F7
0x08	AA-BB-CC-DD-EE-F8	AA-BB-CC-DD-EE-F9	AA-BB-CC-DD-EE-FA	AA-BB-CC-DD-EE-FB
0x0C	AA-BB-CC-DD-EE-FC	AA-BB-CC-DD-EE-FD	AA-BB-CC-DD-EE-FE	AA-BB-CC-DD-EE-FF

Please be noted that all these MAC addresses should be reserved because they are global MAC addresses.

16.3.3 New MBSSID Mode

Since there is MAC address reservation problem in the old MBSSID mode, we provide the new MBSSID mode which will utilize b2 of 6th byte of a virtual MAC address to claim it as locally administered. Address extension would be done on 6th byte.

As to main BSSID, **the b[4:2] of 6th byte** of its MAC address should be:

- Multiple of 2 for 2-BSSID
- Multiple of 4 for 4-BSSID
- Multiple of 8 for 8-BSSID

Taking BssidNum=4 for example:

- ra0: 0**0**:0c:43:00:00:00
- ra1: 0**2**:0c:43:00:00:00 — 0**2** comes from (6th byte 0x00 | b'0000**00**10)
- ra2: 0**6**:0c:43:00:00:00 — 0**6** comes from (6th byte 0x00 | b'0000**01**10)
- ra3: 0**a**:0c:43:00:00:00 — 0**a** comes from (6th byte 0x00 | b'0000**10**10)

16.3.4 Enhanced New MBSSID Mode

The enhanced new MBSSID mode removes the restriction of using the 6th byte since OUI (Consists of 6th, 5th, 4th bytes) is not controllable. Local Administration bit would be turned on and address extension would be done on 3rd byte. This is supported only in new products.

Taking BssidNum=4 for example:

- ra0: 00:0c:43:0**0**:00:00 — 00 is multiple of 4
- ra1: 0**2**:0c:43:0**0**:00:00 — 00 comes from (3rd byte 0x00) + 0
- ra2: 0**2**:0c:43:0**1**:00:00 — 01 comes from (3rd byte 0x00) + 1
- ra3: 0**2**:0c:43:0**2**:00:00 — 02 comes from (3rd byte 0x00) + 2

16.4 Configuration

BssidNum can be configured only through profile and you must restart the interface to make it to work. Other parameters can be configured dynamically through iwpriv command per interface. MBSSID-supported parameters are SSID, AuthMode, EncrypType, WPAPSK, etc.

16.4.1 Example

BssidNum=4
SSID=SSID_A;SSID_B;SSID_C;SSID_D
AuthMode=OPEN;SHARED;WPAPSK;WPA2PSK
EncrypType=NONE;WEP;TKIP;AES

17 Concurrent A+G Settings

Below table is brief example for two interfaces.
For example, Linux HotPlug system found new device would create one driver instance (create new space for driver image) for new device to hold private information (memory consumed).

Interface Bring Up Sequence						
NIC#	Sequence	Normal	WDS			
			1	2	3	4
Two	ifconfig ra0 up	ra0	wds0	wds1	wds2	wds3
	ifconfig ra1 up	ra1	wds4	wds5	wds6	wds7

NIC#	Sequence	Normal	MBSSID			WDS			
						1	2	3	4
Two	ifconfig ra0 up	ra0	ra2	ra3	ra4	wds0	wds1	wds2	wds3
	ifconfig ra1 up	ra1	ra5	ra6	ra7	wds4	wds5	wds6	wds7

WDS IS A VIRTUAL INTERFACE WITHOUT IOCTL FUNCTIONALITY.

18 SNMP MIBs Support List

18.1 RT2860AP Supported v.s. IEEE802dot11-MIB

IEEE802dot11-MIB	Access	Support	OID	RT2860AP.dat
ieee802dot11				
dot11smt		-		
dot11StationConfigTable	not-accessible	-		
dot11StationConfigEntry	not-accessible	-		
dot11StationID	read-write	Y	OID_802_3_CURRENT_ADDRESS	N
dot11MediumOccupancyLimit	read-write	N		N
dot11CFPollable	read-only	N		N
dot11CFPPeriod	read-write	N		N
dot11CFPMaxDuration	read-write	N		N
dot11AuthenticationResponseTimeOut	read-write	N		N
dot11PrivacyOptionImplemented	read-only	Y	RT_OID_802_11_PRIVACYOPTIONIMPLEMENTED	N
dot11PowerManagementMode	read-write	Y	RT_OID_802_11_POWERMANAGEMENTMODE	N
dot11DesiredSSID	read-write	N		N
dot11DesiredBSSType	read-write	N		N
dot11OperationalRateSet	read-write	N		N
dot11BeaconPeriod	read-write	N		N
dot11DTIMPeriod	read-write	N		N
dot11AssociationResponseTimeOut	read-write	N		N
dot11DisassociateReason	read-only	N		N
dot11DisassociateStation	read-only	N		N
dot11DeauthenticateReason	read-only	N		N
dot11DeauthenticateStation	read-only	N		N
dot11AuthenticateFailStatus	read-only	N		N
dot11AuthenticateFailStation	read-only	N		N
dot11AuthenticationAlgorithmsTable	not-accessible	-		-
dot11AuthenticationAlgorithmsEntry	not-	-		-

	access ible			
dot11AuthenticationAlgorithmsIndex	not-access ible	Y		N
dot11AuthenticationAlgorithm	read-only	Y		N
dot11AuthenticationAlgorithmsEnable	read-write	Y		N
dot11WEPDefaultKeysTable	not-access ible	-		-
dot11WEPDefaultKeysEntry	not-access ible	-		-
dot11WEPDefaultKeyIndex	not-access ible	Y		N
dot11WEPDefaultKeyValue	read-write	Y	OID_802_11_WEPDEFAULTKEYVALUE	Y
dot11WEPKeyMappingsTable	not-access ible	-		-
dot11WEPKeyMappingsEntry	not-access ible	-		-
dot11WEPKeyMappingIndex	not-access ible	N		N
dot11WEPKeyMappingAddress	read-create	N		N
dot11WEPKeyMappingWEPOn	read-create	N		N
dot11WEPKeyMappingValue	read-create	N		N
dot11WEPKeyMappingStatus	read-create	N		N
dot11PrivacyTable	not-access ible	-		
dot11PrivacyEntry	not-access ible	-		
dot11PrivacyInvoked	read-write	Y		N
dot11WEPDefaultKeyID	read-write	Y	OID_802_11_WEPDEFAULTKEYID	Y
dot11WEPKeyMappingLength	read-write	Y	RT_OID_802_11_WEPKEYMAPPINGLENGTH	N
dot11ExcludeUnencrypted	read-write	N		N
dot11WEPICVErrorCount	read-only	N		N
dot11WEPExcludedCount	read-only	N		N
dot11SMTnotification	-	-		
dot11Disassociate	-	N		N
dot11Deauthenticate	-	N		N
dot11AuthenticateFail	-	N		N
dot11mac				
dot11OperationTable	not-access	-		

	ible			
dot11OperationEntry	not-access ible	-		
dot11MACAddress	read-only	Y	RT_OID_802_11_MAC_ADDRESS	N
dot11RTSThreshold	read-write	Y	OID_802_11_RTS_THRESHOLD	Y
dot11ShortRetryLimit	read-write	Y	OID_802_11_SHORTRETRYLIMIT	N
dot11LongRetryLimit	read-write	Y	OID_802_11_LONGRETRYLIMIT	N
dot11FragmentationThreshold	read-write	Y	OID_802_11_FRAGMENTATION_THRESHOL D	Y
dot11MaxTransmitMSDULifetime	read-write	N		N
dot11MaxReceiveLifetime	read-write	N		N
dot11ManufacturerID	read-only	Y	RT_OID_802_11_MANUFACTUREID	N
dot11ProductID	read-only	Y	RT_OID_802_11_PRODUCTID	N
dot11CountersTable	not-access ible	-		
dot11CountersEntry	not-access ible	-		
dot11TransmittedFragmentCount	read-only	Y	OID_802_11_STATISTICS	N
dot11MulticastTransmittedFrameCo unt	read-only	Y	OID_802_11_STATISTICS	N
dot11FailedCount	read-only	Y	OID_802_11_STATISTICS	N
dot11RetryCount	read-only	Y	OID_802_11_STATISTICS	N
dot11MultipleRetryCount	read-only	Y	OID_802_11_STATISTICS	N
dot11FrameDuplicateCount	read-only	Y	OID_802_11_STATISTICS	N
dot11RTSSuccessCount	read-only	Y	OID_802_11_STATISTICS	N
dot11RTSFailureCount	read-only	Y	OID_802_11_STATISTICS	N
dot11ACKFailureCount	read-only	Y	OID_802_11_STATISTICS	N
dot11ReceivedFragmentCount	read-only	Y	OID_802_11_STATISTICS	N
dot11MulticastReceivedFrameCount	read-only	Y	OID_802_11_STATISTICS	N
dot11FCSErrorCount	read-only	Y	OID_802_11_STATISTICS	N
dot11TransmittedFrameCount	read-only	N		N
dot11WEPUndecryptableCount	read-only	N		N
dot11GroupAddressesTable	not-access ible	-		-
dot11GroupAddressesEntry	not-access ible	-		-

dot11GroupAddressesIndex	not-accessible	N		N
dot11Address	read-create	N		N
dot11GroupAddressesStatus	read-create	N		N
dot11res				
dot11resAttribute				
dot11ResourceTypeIDName	read-only	-		
dot11ResourceInfoTable	not-accessible	-		
dot11ResourceInfoEntry	not-accessible	-		
dot11manufacturerOUI	read-only	Y	RT_OID_802_11_MANUFACTUREROUI	N
dot11manufacturerName	read-only	Y	RT_OID_802_11_MANUFACTURERNAME	N
dot11manufacturerProductName	read-only	Y	RT_OID_DEVICE_NAME	N
dot11manufacturerProductVersion	read-only	Y	RT_OID_VERSION_INFO	N
dot11phy				
dot11PhyOperationTable	not-accessible	-		
dot11PhyOperationEntry	not-accessible	-		
dot11PHYType	read-only	Y	RT_OID_802_11_PHY_MODE	N
dot11CurrentRegDomain	read-write	Y		Y
dot11TempType	read-only	N		N
dot11PhyAntennaTable	not-accessible	-		
dot11PhyAntennaEntry	not-accessible	-		
dot11CurrentTxAntenna	read-write	Y	OID_802_11_TX_ANTENNA_SELECTED	N
dot11DiversitySupport	read-only	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11CurrentRxAntenna	read-write	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11PhyTxPowerTable	not-accessible	-		
dot11PhyTxPowerEntry	not-accessible	-		
dot11NumberSupportedPowerLevels	read-only	N		N
dot11TxPowerLevel1	read-only	N		N
dot11TxPowerLevel2	read-only	N		N

dot11TxPowerLevel3	read-only	N		N
dot11TxPowerLevel4	read-only	N		N
dot11TxPowerLevel5	read-only	N		N
dot11TxPowerLevel6	read-only	N		N
dot11TxPowerLevel7	read-only	N		N
dot11TxPowerLevel8	read-only	N		N
dot11CurrentTxPowerLevel	read-write	N		N
dot11PhyFHSSTable	not-accessible	-		
dot11PhyFHSSEntry	not-accessible	-		
dot11HopTime	read-only	N		N
dot11CurrentChannelNumber	read-write	N		N
dot11MaxDwellTime	read-only	N		N
dot11CurrentDwellTime	read-write	N		N
dot11CurrentSet	read-write	N		N
dot11CurrentPattern	read-write	N		N
dot11CurrentIndex	read-write	N		N
dot11PhyDSSSTable	not-accessible	-		
dot11PhyDSSSEntry	not-accessible	-		
dot11CurrentChannel	read-write	Y	OID_802_11_CURRENTCHANNEL	Y
dot11CCAModeSupported	read-only	N		N
dot11CurrentCCAMode	read-write	N		N
dot11EDThreshold	read-write	N		N
dot11PhyIRTable	not-accessible	-		
dot11PhyIREntry	not-accessible	-		
dot11CCAWatchdogTimerMax	read-write	N		N
dot11CCAWatchdogCountMax	read-write	N		N
dot11CCAWatchdogTimerMin	read-write	N		N
dot11CCAWatchdogCountMin	read-write	N		N

dot11RegDomainsSupportedTable	not-accessible	-		
dot11RegDomainsSupportEntry	not-accessible	-		
dot11RegDomainsSupportIndex	not-accessible	Y		N
dot11RegDomainsSupportValue	read-only	Y		N
dot11AntennasListTable	not-accessible	-		
dot11AntennasListEntry	not-accessible	-		
dot11AntennaListIndex	not-accessible	Y		N
dot11SupportedTxAntenna	read-write	Y	OID_802_11_TX_ANTENNA_SELECTED	N
dot11SupportedRxAntenna	read-write	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11DiversitySelectionRx	read-write	Y	OID_802_11_RX_ANTENNA_SELECTED	N
dot11SupportedDataRatesTxTable	not-accessible	-		
dot11SupportedDataRatesTxEntry	not-accessible	-		
dot11SupportedDataRatesTxIndex	not-accessible	Y		N
dot11SupportedDataRatesTxValue	read-only	Y	OID_802_11_DESIRED_RATES	N
dot11SupportedDataRatesRxTable	not-accessible	-		
dot11SupportedDataRatesRxEntry	not-accessible	-		
dot11SupportedDataRatesRxIndex	not-accessible	Y	OID_802_11_DESIRED_RATES	
dot11SupportedDataRatesRxValue	read-only	Y		
dot11PhyOFDMTable	not-accessible	-		
dot11PhyOFDMEntry	not-accessible	-		
dot11CurrentFrequency	read-write	N	OID_802_11_CURRENTCHANNEL	Y
dot11TIThreshold	read-write	N		N
dot11FrequencyBandsSupported	read-only	N		N

18.2 RALINK OID for SNMP MIB

RALINK OID for SNMP		
Value	Name	Structure
0x010B	OID_802_11_NUMBER_OF_ANTENNAS	USHORT numant;
0x010C	OID_802_11_RX_ANTENNA_SELECTED	USHORT whichant;
0x010D	OID_802_11_TX_ANTENNA_SELECTED	USHORT whichant;
0x050C	RT_OID_802_11_PHY_MODE	ULONG linfo;
0x050E	OID_802_11_DESIRED_RATES	typedef UCHAR NDIS_802_11_RATES[NDIS_802_11_LENGTH_RATES]; #define NDIS_802_11_LENGTH_RATES 8
0x0514	OID_802_11_RTS_THRESHOLD	ULONGlinfo;
0x0515	OID_802_11_FRAGMENTATION_THRESHOLD	ULONGlinfo;
0x0607	RT_OID_DEVICE_NAME	char name[128];
0x0608	RT_OID_VERSION_INFO	typedef struct PACKED _RT_VERSION_INFO{ UCHAR DriverVersionW; UCHAR DriverVersionX; UCHAR DriverVersionY; UCHAR DriverVersionZ; UINT DriverBuildYear; UINT DriverBuildMonth; UINT DriverBuildDay; } RT_VERSION_INFO, *PRT_VERSION_INFO;
0x060A	OID_802_3_CURRENT_ADDRESS	char addr[128];
0x060E	OID_802_11_STATISTICS	typedef struct _NDIS_802_11_STATISTICS { ULONG Length; // Length of structure ULONG TransmittedFragmentCount; ULONG MulticastTransmittedFrameCount; ULONG FailedCount; ULONG RetryCount; ULONG MultipleRetryCount; ULONG RTSSuccessCount; ULONG RTSFailureCount; ULONG ACKFailureCount; ULONG FrameDuplicateCount; ULONG ReceivedFragmentCount;

		ULONG MulticastReceivedFrameCount; ULONG FCSErrorCount; } NDIS_802_11_STATISTICS, PNDIS_802_11_STATISTICS;
0x0700	RT_OID_802_11_MANUFACTURER OUI	char oui[128];
0x0701	RT_OID_802_11_MANUFACTURER NAME	char name[128];
0x0702	RT_OID_802_11_RESOURCETYPEI DNAME	char name[128];
0x0703	RT_OID_802_11_PRIVACYOPTIONI MPLEMENTED	ULONG linfo;
0x0704	RT_OID_802_11_POWERMANAGE MENTMODE	ULONG linfo;
0x0705	OID_802_11_WEPDEFAULTKEYVAL UE	typedef struct _DefaultKeyIdxValue { UCHARKeyIdx; UCHARValue[16]; }DefaultKeyIdxValue;
0x0706	OID_802_11_WEPDEFAULTKEYID	UCHARkeyid;
0x0707	RT_OID_802_11_WEPKEYMAPPIN GLENGTH	UCHAR len;
0x0708	OID_802_11_SHORTRETRYLIMIT	ULONGlinfo;
0x0709	OID_802_11_LONGRETRYLIMIT	ULONGlinfo;
0x0710	RT_OID_802_11_PRODUCTID	char id[128];
0x0711	RT_OID_802_11_MANUFACTUREID	char id[128];
0x0712	OID_802_11_CURRENTCHANNEL	UCHAR channel
0x0713	RT_OID_802_11_MAC_ADDRESS	char macaddress[128]

19 IOCTL I/O Control Interface

19.1 Parameters for iwconfig's IOCTL

Access	Description	ID	Parameters
Get	BSSID, MAC Address	SIOCGIFHWADDR	wrq->u.name, (length = 6)
	WLAN Name	SIOCGIWNAME	wrq->u.name = "RT2800 SoftAP", length = strlen(wrq->u.name)
	SSID	SIOCGIWESSID	struct iw_point *erq = &wrq->u.essid; erq->flags=1; erq->length = pAd->PortCfg.MBSSID[pAd->IoctlIF].SsidLen; if(erq->pointer) { if(copy_to_user(erq->pointer, pAd->PortCfg.MBSSID[pAd->IoctlIF].Ssid, erq->length)) { Status = -EFAULT; break; } }
	Channel / Frequency (Hz)	SIOCGIWFREQ	wrq->u.freq.m = pAd->PortCfg.Channel; wrq->u.freq.e = 0; wrq->u.freq.i = 0;
	Bit Rate (bps)	SIOCGIWRATE	wrq->u.bitrate.value = RateIdTo500Kbps[pAd->PortCfg.MBSSID[pAd->IoctlIF].TxRate] * 500000; wrq->u.bitrate.disabled = 0;
	AP's MAC address	SIOCGIWAP	wrq->u.ap_addr.sa_family = ARPHRD_ETHER; memcpy(wrq->u.ap_addr. sa_data, &pAd->PortCfg.MBSSID[pAd->IoctlIF].Bssid, ETH_ALEN);
	Operation Mode	SIOCGIWMODE	wrq->u.mode = IW_MODE_INFRA;
	Range of Parameters	SIOCGIWRANGE	range.we_version_compiled = WIRELESS_EXT; range.we_version_source = 14;
	Scanning Results	SIOCGIWSCAN	typedef struct _NDIS_802_11_SITE_SURVEY_TABLE { LONG Channel; LONG Rssi; UCHAR Ssid[33]; UCHAR Bssid[18]; UCHAR EncrypT[8]; } NDIS_802_11_SITE_SURVEY_TABLE, *PNDIS_802_11_SITE_SURVEY_TABLE; wrq->u.data.length = N* sizeof(NDIS_802_11_SITE_SURVEY_TABLE); copy_to_user(wrq->u.data.pointer, site_survey_table, wrq->u.data.length);
	Client	SIOCGIWAPLIST	typedef struct _NDIS_802_11_STATION_TABLE

	Association List		{ UCHAR MacAddr[18]; ULONG Aid; ULONG PsMode; ULONG LastDataPacketTime; ULONG RxByteCount; ULONG TxByteCount; ULONG CurrTxRate; ULONG LastTxRate; } NDIS_802_11_STATION_TABLE, *PNDIS_802_11_STATION_TABLE; wrq->u.data.length = i * sizeof(NDIS_802_11_STATION_TABLE); copy_to_user(wrq->u.data.pointer, sta_list_table, wrq->u.data.length);
Set	Trigger Scanning	SIOCSIWSCAN	ApSiteSurvey(pAd);

19.2 Parameters for iwpriv's IOCTL

Please refer section 4 and 5 to have iwpriv parameters and values.

Parameters:

```
int     socket_id;
char    name[25];          // interface name
char    data[255];         // command string
struct  iwreq wrq;
```

Default setting:

```
wrq.ifr_name = name = "ra0";        // interface name
wrq.u.data.pointer = data;          // data buffer of command string
wrq.u.data.length = strlen(data);   // length of command string
wrq.u.data.flags = 0;
```

19.2.1 Iwpriv Set DATA

THESE PARAMETERS ARE THE SAME AS IWPRIV

Command and IOCTL Function		
Set Data		
Function Type	Command	IOCTL
RTPRIV_IOCTL_SET	iwpriv ra0 set SSID=RT2800AP	sprintf(name, "ra0"); strcpy(data, "SSID=RT2800AP"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);

19.2.2 Iwpriv Get DATA

THESE PARAMETERS ARE THE SAME AS IWPRIV

Command and IOCTL Function

Get Data		
Function Type	Command	IOCTL
RTPRIV_IOCTL_STATISTICS	Iwpriv ra0 stat	sprintf(name, "ra0"); strcpy(data, "stat"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_STATISTICS, &wrq);
RTPRIV_IOCTL_GSITESURVEY	Iwpriv ra0 get_site_survey	sprintf(name, "ra0"); strcpy(data, "get_site_survey"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_GSITESURVEY, &wrq);
RTPRIV_IOCTL_GET_MAC_TABLE	Iwpriv ra0 get_mac_table	sprintf(name, "ra0"); strcpy(data, "get_mac_table"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_GET_MAC_TABLE, &wrq);
RTPRIV_IOCTL_SHOW	Iwpriv ra0 show	sprintf(name, "ra0"); strcpy(data, "get_mac_table"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_SHOW, &wrq);
RTPRIV_IOCTL_WSC_PROFILE	Iwpriv ra0 get_wsc_profile	sprintf(name, "ra0"); strcpy(data, "get_mac_table"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_WSC_PROFILE, &wrq);
RTPRIV_IOCTL_QUERY_BATABLE	Iwpriv ra0 get_ba_table	sprintf(name, "ra0"); strcpy(data, "get_mac_table"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_QUERY_BATABLE, &wrq);

19.2.3 Iwpriv Set Data: BBP, MAC and EEPROM

Command and IOCTL Function		
Set Data: BBP, MAC and EEPROM, Parameters is Same as iwpriv		
Type	Command	IOCTL
RTPRIV_IOCTL_BBP (Set BBP Register Value)	Iwpriv ra0 bbp 17=32	sprintf(name, "ra0"); strcpy(data, " bbp 17=32"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_BBP, &wrq);
RTPRIV_IOCTL_MAC (Set MAC Register Value)	Iwpriv ra0 mac 3000=12345678	sprintf(name, "ra0"); strcpy(data, " mac 3000=12345678"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_MAC, &wrq);
RTPRIV_IOCTL_E2P (Set EEPROM Value)	Iwpriv ra0 e2p 40=1234	sprintf(name, "ra0"); strcpy(data, " e2p 40=1234"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_E2P, &wrq);

19.2.4 Iwpriv Get Data: BBP, MAC and EEPROM

Command and IOCTL Function		
Get Data: BBP, MAC and EEPROM , Parameters is Same as iwpriv		
Type	Command	IOCTL
RTPRIV_IOCTL_BBP (Get BBP Register Value)	Iwpriv ra0 bbp 17	sprintf(name, "ra0"); strcpy(data, " bbp 17"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_BBP, &wrq);
RTPRIV_IOCTL_MAC (Get MAC Register Value)	Iwpriv ra0 mac 3000	sprintf(name, "ra0"); strcpy(data, " mac 3000"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_MAC, &wrq);
RTPRIV_IOCTL_E2P	Iwpriv ra0 e2p 40	sprintf(name, "ra0");

(Get EEPROM Value)		strcpy(data, " e2p 40"); strcpy(wrq.ifr_name, name); wrq.u.data.length = strlen(data); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_E2P, &wrq);

19.2.5 Iwpriv Set Raw Data

IOCTL Function	
Set Raw Data by I/O Control Interface	
Function Type	IOCTL
RTPRIV_IOCTL_RADIUS_DATA	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0x55, 100); wrq.u.data.length = 100; wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_RADIUS_DATA, &wrq);
RTPRIV_IOCTL_ADD_WPA_KEY	NDIS_802_11_KEY *vp; sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(NDIS_802_11_KEY)); vp = (NDIS_802_11_KEY *)&data; vp->Length = sizeof(NDIS_802_11_KEY); memset(vp->addr, 0x11, 6); vp->KeyIndex = 2; vp->KeyLength = 32; memset(vp->KeyMaterial, 0xAA, 32); wrq.u.data.length = sizeof(NDIS_802_11_KEY); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_ADD_WPA_KEY, &wrq);
RTPRIV_IOCTL_ADD_PMKID_CACHE	NDIS_802_11_KEY *vp; sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(NDIS_802_11_KEY)); vp = (NDIS_802_11_KEY *)&data; vp->Length = sizeof(NDIS_802_11_KEY); memset(vp->addr, 0x11, 6); vp->KeyIndex = 2; vp->KeyLength = 32; memset(vp->KeyMaterial, 0xBB, 32); wrq.u.data.length = sizeof(NDIS_802_11_KEY); wrq.u.data.pointer = data; wrq.u.data.flags = 0; ioctl(socket_id, RTPRIV_IOCTL_ADD_PMKID_CACHE, &wrq);

19.2.6 Set Raw Data with Flags

IOCTL Function	
Set Raw Data by I/O Control Interface with Flags	
Function Type	IOCTL
RT_SET_APD_PID	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, 4); data[0] = 12; wrq.u.data.length = 4; wrq.u.data.pointer = data; wrq.u.data.flags = RT_SET_APD_PID; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_SET_DEL_MAC_ENTRY	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0xdd, 6); strcpy(wrq.ifr_name, name); wrq.u.data.length = 6; wrq.u.data.pointer = data; wrq.u.data.flags = RT_SET_DEL_MAC_ENTRY; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_WSC_SET_SELECTED_REGISTRAR	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, decodeStr, decodeLen); strcpy(wrq.ifr_name, name); wrq.u.data.length = decodeLen; wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_WSC_SET_SELECTED_REGISTRAR; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_WSC_EAPMSG	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, wscU2KMsg, wscU2KMsgLen); strcpy(wrq.ifr_name, name); wrq.u.data.length = wscU2KMsgLen; wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_WSC_EAPMSG; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

19.2.7 Get Raw Data with Flags

IOCTL Function	
Get Raw Data by I/O Control Interface with Flags	
Function Type	IOCTL
RT_QUERY_ATE_TXDONE_COUNT	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(ULONG)); wrq.u.data.length = sizeof(ULONG); wrq.u.data.pointer = data;

	wrq.u.data.flags = RT_QUERY_ATE_TXDONE_COUNT; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_QUERY_SIGNAL_CONTEXT	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(RT_SIGNAL_STRUC)); strcpy(wrq.ifr_name, name); wrq.u.data.length = sizeof(RT_SIGNAL_STRUC); wrq.u.data.pointer = data; wrq.u.data.flags = RT_QUERY_SIGNAL_CONTEXT; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_WSC_QUERY_STATUS	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(INT)); strcpy(wrq.ifr_name, name); wrq.u.data.length = sizeof(INT); wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_WSC_QUERY_STATUS; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_WSC_PIN_CODE	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(ULONG)); strcpy(wrq.ifr_name, name); wrq.u.data.length = sizeof(ULONG); wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_WSC_PIN_CODE; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_WSC_UUID	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(UCHAR)); strcpy(wrq.ifr_name, name); wrq.u.data.length = sizeof(UCHAR); wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_WSC_UUID; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_WSC_MAC_ADDRESS	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, MAC_ADDR_LEN); strcpy(wrq.ifr_name, name); wrq.u.data.length = MAC_ADDR_LEN; wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_WSC_MAC_ADDRESS; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_GET_PHY_MODE	sprintf(name, "ra0"); strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(ULONG)); strcpy(wrq.ifr_name, name); wrq.u.data.length = sizeof(ULONG); wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_GET_PHY_MODE; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
RT_OID_GET_LLTD_ASSO_TANLE	sprintf(name, "ra0");

	strcpy(wrq.ifr_name, name); memset(data, 0, sizeof(RT_LLTD_ASSOICATION_TABLE)); strcpy(wrq.ifr_name, name); wrq.u.data.length = sizeof(RT_LLTD_ASSOICATION_TABLE); wrq.u.data.pointer = data; wrq.u.data.flags = RT_OID_GET_LLTD_ASSO_TANLE; ioctl(socket_id, RT_PRIV_IOCTL, &wrq);

19.3 Sample user space Applications

```
//===============================================================
//
// rtuser:
//        1. User space application to demo how to use IOCTL function.
//        2. Most of the IOCTL function is defined as "CHAR" type and return with string message.
//        3. Use sscanf to get the raw data back from string message.
//        4. The command format "parameter=value" is same as iwpriv command format.
//        5. Remember to insert driver module and bring interface up prior execute rtuser.
//                change folder path to driver "Module"
//                dos2unix *                ; in case the files are modified from other OS environment
//                chmod 644 *
//                chmod 755 Configure
//                make config
//                make
//                insmod RT2800ap.o
//                ifconfig ra0 up
//
// Refer Linux/if.h to have
//                #define ifr_name    ifr_ifrn.ifrn_name                /* interface name    */
//
// Make:
//                cc -Wall -ortuser rtuser.c
//
// Run:
//                ./rtuser
//
//===============================================================

#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <unistd.h>                /* for close */
#include <Linux/wireless.h>

//=====================================================================

#if WIRELESS_EXT <= 11
#ifndef SIOCDEVPRIVATE
#define SIOCDEVPRIVATE                0x8BE0
#endif
#define SIOCIWFIRSTPRIV               SIOCDEVPRIVATE
#endif

//
//SET/GET CONVENTION :
// * -----------------
// * Simplistic summary :
```

```
// *         o even numbered ioctls are SET, restricted to root, and should not
// *    return arguments (get_args = 0).
// *         o odd numbered ioctls are GET, authorised to anybody, and should
// *    not expect any arguments (set_args = 0).
//
#define RT_PRIV_IOCTL                   (SIOCIWFIRSTPRIV + 0x01)
#define RTPRIV_IOCTL_SET                                (SIOCIWFIRSTPRIV + 0x02)
#define RTPRIV_IOCTL_BBP                (SIOCIWFIRSTPRIV + 0x03)
#define RTPRIV_IOCTL_MAC                (SIOCIWFIRSTPRIV + 0x05)
#define RTPRIV_IOCTL_E2P                (SIOCIWFIRSTPRIV + 0x07)
#define RTPRIV_IOCTL_STATISTICS                 (SIOCIWFIRSTPRIV + 0x09)
#define RTPRIV_IOCTL_ADD_PMKID_CACHE    (SIOCIWFIRSTPRIV + 0x0A)
#define RTPRIV_IOCTL_RADIUS_DATA        (SIOCIWFIRSTPRIV + 0x0C)
#define RTPRIV_IOCTL_GSITESURVEY                        (SIOCIWFIRSTPRIV + 0x0D)
#define RTPRIV_IOCTL_ADD_WPA_KEY        (SIOCIWFIRSTPRIV + 0x0E)
#define RTPRIV_IOCTL_GET_MAC_TABLE                      (SIOCIWFIRSTPRIV + 0x0F)


#define OID_GET_SET_TOGGLE                              0x8000

#define RT_QUERY_ATE_TXDONE_COUNT                       0x0401
#define RT_QUERY_SIGNAL_CONTEXT                         0x0402
#define RT_SET_APD_PID                                  (OID_GET_SET_TOGGLE + 0x0405)
#define RT_SET_DEL_MAC_ENTRY                            (OID_GET_SET_TOGGLE + 0x0406)

//---------------------------------------------------------

#ifndef     TRUE
#define     TRUE            1
#endif

#ifndef     FALSE
#define     FALSE   0
#endif

#define MAC_ADDR_LEN                                    6
#define ETH_LENGTH_OF_ADDRESS                           6
#define MAX_LEN_OF_MAC_TABLE                            64

//---------------------------------------------------------

typedef struct _COUNTERS
{
        unsigned long       TxSuccessTotal;;
        unsigned long       TxSuccessWithRetry;
        unsigned long       TxFailWithRetry;
        unsigned long       RtsSuccess;
        unsigned long       RtsFail;
        unsigned long       RxSuccess;
        unsigned long       RxWithCRC;
        unsigned long       RxDropNoBuffer;
        unsigned long       RxDuplicateFrame;
        unsigned long       FalseCCA;
        unsigned long       RssiA;
        unsigned long       RssiB;
}       COUNTERS;
```

PS. User can check with "iwpriv ra0 stat" to make sure the TXRX status is correct when porting the ATE related test program.

```
//---------------------------------------------------------
```

```
typedef    struct _SITE_SURVEY
{
           unsigned char          channel;
           unsigned short         rssi;
           unsigned char          ssid[33];
           unsigned char          bssid[6];
           unsigned char          security[9];
}          SITE_SURVEY;

//---------------------------------------------------------

typedef union  _MACHTTRANSMIT_SETTING {
           struct     {
           unsigned short         MCS:7;          // MCS
           unsigned short         BW:1;                    //channel bandwidth 20MHz or 40 MHz
           unsigned short         ShortGI:1;
           unsigned short         STBC:2;         //SPACE
           unsigned short         rsv:3;
           unsigned short         MODE:2;         // Use definition MODE_xxx.
           }          field;
           unsigned short         word;
 } MACHTTRANSMIT_SETTING, *PMACHTTRANSMIT_SETTING;

typedef struct _RT_802_11_MAC_ENTRY {
   unsigned char                  Addr[6];
   unsigned char                  Aid;
   unsigned char                  Psm;                     // 0:PWR_ACTIVE, 1:PWR_SAVE
   unsigned char                  MimoPs;         // 0:MMPS_STATIC, 1:MMPS_DYNAMIC, 3:MMPS_Enabled
   MACHTTRANSMIT_SETTING          TxRate;
} RT_802_11_MAC_ENTRY, *PRT_802_11_MAC_ENTRY;

typedef struct _RT_802_11_MAC_TABLE {
   unsigned long                  Num;
   RT_802_11_MAC_ENTRY Entry[MAX_LEN_OF_MAC_TABLE];
} RT_802_11_MAC_TABLE, *PRT_802_11_MAC_TABLE;

// Key mapping keys require a BSSID
typedef struct _NDIS_802_11_KEY
{
           unsigned long          Length;      // Length of this structure
           unsigned char          addr[6];
           unsigned long          KeyIndex;
           unsigned long          KeyLength;     // length of key in bytes
           unsigned char          KeyMaterial[32];  // variable length depending on above field
} NDIS_802_11_KEY, *PNDIS_802_11_KEY;

typedef struct _RT_SIGNAL_STRUC {
           unsigned short         Sequence;
           unsigned char          MacAddr[MAC_ADDR_LEN];
           unsigned char          CurrAPAddr[MAC_ADDR_LEN];
           unsigned char          Sig;
} RT_SIGNAL_STRUC, *PRT_SIGNAL_STRUC;

//---------------------------------------------------------

COUNTERS             counter;
SITE_SURVEY    SiteSurvey[100];
char                 data[4096];

//=========================================================================
```

```
int main( int argc, char ** argv )
{
        char            name[25];
        int             socket_id;
        struct   iwreq wrq;
        int             ret;

        // open socket based on address family: AF_NET ----------------------------
        socket_id = socket(AF_INET, SOCK_DGRAM, 0);
        if(socket_id < 0)
        {
                printf("\nrtuser::error::Open socket error!\n\n");
                return -1;
        }

        // set interface name as "ra0" --------------------------------------------
        sprintf(name, "ra0");
        memset(data, 0x00, 255);
//
//example of iwconfig ioctl function ==========================================
//
        // get wireless name -----------------------------------------------------
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = 255;
        wrq.u.data.pointer = data;
        wrq.u.data.flags = 0;
        ret = ioctl(socket_id, SIOCGIWNAME, &wrq);
        if(ret != 0)
        {
                printf("\nrtuser::error::get wireless name\n\n");
                goto rtuser_exit;
        }

        printf("\nrtuser[%s]:%s\n", name, wrq.u.name);
//
//example of iwpriv ioctl function ============================================
//
        //WPAPSK, remove "set" string --------------------------------------------
        memset(data, 0x00, 255);
        strcpy(data, "WPAPSK=11223344");
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = strlen(data)+1;
        wrq.u.data.pointer = data;
        wrq.u.data.flags = 0;
        ret = ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);
        if(ret != 0)
        {
                printf("\nrtuser::error::set wpapsk\n\n");
                goto rtuser_exit;
        }
        //set e2p, remove "e2p" string -------------------------------------------
        memset(data, 0x00, 255);
        strcpy(data, "80=1234");
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = strlen(data)+1;
        wrq.u.data.pointer = data;
        wrq.u.data.flags = 0;
        ret = ioctl(socket_id, RTPRIV_IOCTL_E2P, &wrq);
        if(ret != 0)
```

```
{
        printf("\nrtuser::error::set eeprom\n\n");
        goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
        int addr, value, p1;

        // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
        sscanf(wrq.u.data.pointer, "\n[%dx%02X]:%04X  ", &p1, &addr, &value);
        printf("\nSet EEP[0x%02X]:0x%04X\n", addr, value);
}

//get e2p, remove "e2p" string -------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "80");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_E2P, &wrq);
if(ret != 0)
{
        printf("\nrtuser::error::get eeprom\n\n");
        goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
        int addr, value, p1, p2;

        // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
        sscanf(wrq.u.data.pointer, "\n[%dx%04X]:%dx%X  ", &p1, &addr, &p2, &value);
        printf("\nGet EEP[0x%02X]:0x%04X\n", addr, value);
}

//set mac, remove "mac" string -------------------------------------------
memset(data, 0x00, 255);
strcpy(data, "2b4f=1");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_MAC, &wrq);
if(ret != 0)
{
        printf("\nrtuser::error::set mac register\n\n");
        goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
        int addr, value, p1;

        // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
        sscanf(wrq.u.data.pointer, "\n[%dx%08X]:%08X  ", &p1, &addr, &value);
        printf("\nSet MAC[0x%08X]:0x%08X\n", addr, value);
}
```

//get mac, remove "mac" string ---

```
memset(data, 0x00, 255);
strcpy(data, "2b4f");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_MAC, &wrq);
if(ret != 0)
{
        printf("\nrtuser::error::get mac register\n\n");
        goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
        int addr, value, p1;

        // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
        sscanf(wrq.u.data.pointer, "\n[%dx%08X]:%08X  ", &p1, &addr, &value);
        printf("\nGet MAC[0x%08X]:0x%08X\n", addr, value);
}
```

//set bbp, remove "bbp" string ---

```
memset(data, 0x00, 255);
strcpy(data, "17=32");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_BBP, &wrq);
if(ret != 0)
{
        printf("\nrtuser::error::set bbp register\n\n");
        goto rtuser_exit;
}

//printf("\n%s\n", wrq.u.data.pointer);
{
        int id, addr, value, p1;

        // string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
        sscanf(wrq.u.data.pointer, "\nR%02d[%dx%02X]:%02X\n", &id, &p1, &addr, &value);
        printf("\nSet BBP R%02d[0x%02X]:0x%02X\n", id, addr, value);
}
```

//get bbp, remove "bbp" string ---

```
memset(data, 0x00, 255);
strcpy(data, "17");
strcpy(wrq.ifr_name, name);
wrq.u.data.length = strlen(data)+1;
wrq.u.data.pointer = data;
wrq.u.data.flags = 0;
ret = ioctl(socket_id, RTPRIV_IOCTL_BBP, &wrq);
if(ret != 0)
{
        printf("\nrtuser::error::get bbp register\n\n");
        goto rtuser_exit;
}
```

```
	//printf("\n%s\n", wrq.u.data.pointer);
	{
		int id, addr, value, p1;

		// string format: "\n[0x%02X]:0x%04X  " ==> "[0x20]:0x0C02"
		sscanf(wrq.u.data.pointer, "\nR%02d[%dx%02X]:%02X  ", &id, &p1, &addr, &value);
		printf("\nGet BBP R%02d[0x%02X]:0x%02X\n", id, addr, value);
	}

	//get statistics, remove "stat" string -----------------------------------
	memset(data, 0x00, 2048);
	strcpy(data, "");
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = 0;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ret = ioctl(socket_id, RTPRIV_IOCTL_STATISTICS, &wrq);
	if(ret != 0)
	{
		printf("\nrtuser::error::get statistics\n\n");
		goto rtuser_exit;
	}

	printf("\n========= Get AP Statistics ==========\n");
	{
		int i;
		char *sp = wrq.u.data.pointer;
		unsigned long *cp = (unsigned long *)&counter;

		for (i = 0 ; i < 13 ; i++)
		{
			sp = strstr(sp, "= ");
			sp = sp+2;
			sscanf(sp, "%ul", (unsigned int *)&cp[i]);
		}
	  printf("Tx success                              = %u\n", (unsigned int)counter.TxSuccessTotal);
	  printf("Tx success without retry                = %u\n", (unsigned int)

	counter.TxSuccessWithoutRetry);
	  printf("Tx success after retry          = %u\n", (unsigned int)counter.TxSuccessWithRetry);
	  printf("Tx fail to Rcv ACK after retry  = %u\n", (unsigned int)counter.TxFailWithRetry);
	  printf("RTS Success Rcv CTS                     = %u\n", (unsigned int)counter.RtsSuccess);
	  printf("RTS Fail Rcv CTS                = %u\n", (unsigned int)counter.RtsFail);
	  printf("Rx success                              = %u\n", (unsigned int)counter.RxSuccess);
	  printf("Rx with CRC                             = %u\n", (unsigned int)counter.RxWithCRC);
	  printf("Rx drop due to out of resource= %u\n", (unsigned int)counter.RxDropNoBuffer);
	  printf("Rx duplicate frame              = %u\n", (unsigned int)counter.RxDuplicateFrame);
	  printf("False CCA (one second)          = %u\n", (unsigned int)counter.FalseCCA);
	  printf("RSSI-A                                  = %d\n", (  signed int)counter.RssiA);
	  printf("RSSI-B (if available)           = %d\n", (  signed int)counter.RssiB);
	}

#if 0
	//set AP to do site survey, remove "set" string --------------------------
	memset(data, 0x00, 255);
	strcpy(data, "SiteSurvey=1");
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = strlen(data)+1;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
```

```
	ret = ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);
#endif

	//get AP's site survey, remove "get_site_survey" string -------------------
	memset(data, 0x00, 2048);
	strcpy(data, "");
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = 4096;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ret = ioctl(socket_id, RTPRIV_IOCTL_GSITESURVEY, &wrq);
	if(ret != 0)
	{
		printf("\nrtuser::error::get site survey\n\n");
		goto rtuser_exit;
	}

	//printf("\n%s\n", wrq.u.data.pointer);
	printf("\n========== Get Site Survey AP List ==========");
	if(wrq.u.data.length > 0)
	{
		int	i, apCount;
		char *sp, *op;
		int	len = wrq.u.data.length;

		op = sp = wrq.u.data.pointer;
		sp = sp+1+8+8+35+19+8+1;
		i = 0;
		// santy check
		//	1. valid char data
		//	2. rest length is larger than per line length ==> (1+8+8+35+19+8+1)
		while(*sp && ((len - (sp-op)) > (1+8+8+35+19+8)))
		{
			//if(*sp++ == '\n')
			//	continue;
			//printf("\n\nAP Count: %d\n", i);

			sscanf(sp, "%d", (int *)&SiteSurvey[i].channel);
			//printf("channel: %d\n", SiteSurvey[i].channel);

			sp = strstr(sp, "-");
			sscanf(sp, "-%d", (int *)&SiteSurvey[i].rssi);
			//printf("rssi: -%d\n", SiteSurvey[i].rssi);

			sp = sp+8;
			strncpy((char *)&SiteSurvey[i].ssid, sp, 32);
			SiteSurvey[i].ssid[32] = '\0';
			//printf("ssid: %s\n", SiteSurvey[i].ssid);

			sp = sp+35;
			sscanf(sp, "%02x:%02x:%02x:%02x:%02x:%02x",
					(int *)&SiteSurvey[i].bssid[0], (int *)&SiteSurvey[i].bssid[1],
					(int *)&SiteSurvey[i].bssid[2], (int *)&SiteSurvey[i].bssid[3],
					(int *)&SiteSurvey[i].bssid[4], (int *)&SiteSurvey[i].bssid[5]);
			//printf("bssid: %02x:%02x:%02x:%02x:%02x:%02x\n",
			//		SiteSurvey[i].bssid[0], SiteSurvey[i].bssid[1],
			//		SiteSurvey[i].bssid[2], SiteSurvey[i].bssid[3],
			//		SiteSurvey[i].bssid[4], SiteSurvey[i].bssid[5]);

			sp = sp+19;
```

```
			strncpy((char *)&SiteSurvey[i].security, sp, 8);
			SiteSurvey[i].security[8] = '\0';
			//printf("security: %s\n", SiteSurvey[i].security);

			sp = sp+8+1;
			i = i+1;
		}

		apCount = i;
		printf("\n%-4s%-8s%-8s%-35s%-20s%-8s\n",
			"AP", "Channel", "RSSI", "SSID", "BSSID", "Security");
		for(i = 0 ; i < apCount ; i++)
		{//4+8+8+35+20+8
			printf("%-4d", i+1);
			printf("%-8d", SiteSurvey[i].channel);
			printf("-%-7d", SiteSurvey[i].rssi);
			printf("%-35s", SiteSurvey[i].ssid);
			printf("%02X:%02X:%02X:%02X:%02X:%02X   ",
					SiteSurvey[i].bssid[0], SiteSurvey[i].bssid[1],
					SiteSurvey[i].bssid[2], SiteSurvey[i].bssid[3],
					SiteSurvey[i].bssid[4], SiteSurvey[i].bssid[5]);
			printf("%-8s\n", SiteSurvey[i].security);
		}
	}
```

//get AP's mac table, remove "get_mac_table" string ----------------------

```
	memset(data, 0x00, 2048);
	strcpy(data, "");
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = 2048;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ret = ioctl(socket_id, RTPRIV_IOCTL_GET_MAC_TABLE, &wrq);
	if(ret != 0)
	{
		printf("\nrtuser::error::get mac table\n\n");
		goto rtuser_exit;
	}

	printf("\n========== Get Associated MAC Table ==========");
	{
		RT_802_11_MAC_TABLE			*mp;
		int				i;

		mp = (RT_802_11_MAC_TABLE *)wrq.u.data.pointer;
		printf("\n%-4s%-20s%-4s%-10s%-10s%-10s\n",
			"AID", "MAC_Address", "PSM", "LastTime", "RxByte", "TxByte");

		for(i = 0 ; i < mp->Num ; i++)
		{
			printf("%-4d", mp->Entry[i].Aid);
			printf("%02X:%02X:%02X:%02X:%02X:%02X   ",
					mp->Entry[i].Addr[0], mp->Entry[i].Addr[1],
					mp->Entry[i].Addr[2], mp->Entry[i].Addr[3],
					mp->Entry[i].Addr[4], mp->Entry[i].Addr[5]);
			printf("%-4d", mp->Entry[i].Psm);
			printf("%-10u", (unsigned int)mp->Entry[i].HSCounter.LastDataPacketTime);
			printf("%-10u", (unsigned int)mp->Entry[i].HSCounter.TotalRxByteCount);
			printf("%-10u", (unsigned int)mp->Entry[i].HSCounter.TotalTxByteCount);
			printf("\n");
```

```
		}
		printf("\n");
	}

//set: raw data
//	RTPRIV_IOCTL_RADIUS_DATA
//	RTPRIV_IOCTL_ADD_WPA_KEY
//	RTPRIV_IOCTL_ADD_PMKID_CACHE

	//set RADIUS Data -------------------------------------------------------
	printf("\nrtuser::set radius data\n\n");
	memset(data, 0x55, 100);
	strcpy(wrq.ifr_name, name);
	wrq.u.data.length = 100;
	wrq.u.data.pointer = data;
	wrq.u.data.flags = 0;
	ret = ioctl(socket_id, RTPRIV_IOCTL_RADIUS_DATA, &wrq);
	if(ret != 0)
	{
		printf("\nrtuser::error::set radius data\n\n");
		goto rtuser_exit;
	}

	//add WPA Key -----------------------------------------------------------
	printf("\nrtuser::add wpa key\n\n");
	{
		NDIS_802_11_KEY		*vp;

		memset(data, 0, sizeof(NDIS_802_11_KEY));
		vp = (NDIS_802_11_KEY *)&data;

		vp->Length = sizeof(NDIS_802_11_KEY);
		memset(vp->addr, 0x11, 6);
		vp->KeyIndex = 2;
		vp->KeyLength = 32;
		memset(vp->KeyMaterial, 0xAA, 32);

		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = sizeof(NDIS_802_11_KEY);
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ret = ioctl(socket_id, RTPRIV_IOCTL_ADD_WPA_KEY, &wrq);
		if(ret != 0)
		{
			printf("\nrtuser::error::add wpa key\n\n");
			goto rtuser_exit;
		}
	}

	//add PMKID_CACHE -------------------------------------------------------
	printf("\nrtuser::add PMKID_CACHE\n\n");
	{
		NDIS_802_11_KEY		*vp;

		memset(data, 0, sizeof(NDIS_802_11_KEY));
		vp = (NDIS_802_11_KEY *)&data;

		vp->Length = sizeof(NDIS_802_11_KEY);
		memset(vp->addr, 0x11, 6);
		vp->KeyIndex = 2;
```

```
                vp->KeyLength = 32;
                memset(vp->KeyMaterial, 0xBB, 32);

                strcpy(wrq.ifr_name, name);
                wrq.u.data.length = sizeof(NDIS_802_11_KEY);
                wrq.u.data.pointer = data;
                wrq.u.data.flags = 0;
                ret = ioctl(socket_id, RTPRIV_IOCTL_ADD_PMKID_CACHE, &wrq);
                if(ret != 0)
                {
                        printf("\nrtuser::error::add PMKID_CACHE\n\n");
                        goto rtuser_exit;
                }
        }

//set: raw data
//      RT_SET_APD_PID
//      RT_SET_DEL_MAC_ENTRY

        //set APD_PID ------------------------------------------------------------
        printf("\nrtuser::set APD_PID\n\n");
        memset(data, 0, 4);
        data[0] = 12;
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = 4;
        wrq.u.data.pointer = data;
        wrq.u.data.flags = RT_SET_APD_PID;
        ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
        if(ret != 0)
        {
                printf("\nrtuser::error::set APD_PID\n\n");
                goto rtuser_exit;
        }

        //set DEL_MAC_ENTRY ------------------------------------------------------
        printf("\nrtuser::set DEL_MAC_ENTRY\n\n");
        memset(data, 0xdd, 6);
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = 6;
        wrq.u.data.pointer = data;
        wrq.u.data.flags = RT_SET_DEL_MAC_ENTRY;
        ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
        if(ret != 0)
        {
                printf("\nrtuser::error::set DEL_MAC_ENTRY\n\n");
                goto rtuser_exit;
        }

//get: raw data
//      RT_QUERY_ATE_TXDONE_COUNT
//      RT_QUERY_SIGNAL_CONTEXT

        //get ATE_TXDONE_COUNT ---------------------------------------------------
        printf("\nrtuser::get ATE_TXDONE_COUNT\n\n");
        memset(data, 0, 4);
        strcpy(wrq.ifr_name, name);
        wrq.u.data.length = 4;
        wrq.u.data.pointer = data;
        wrq.u.data.flags = RT_QUERY_ATE_TXDONE_COUNT;
        ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
```

```
		if(ret != 0)
		{
			printf("\nrtuser::error::get ATE_TXDONE_COUNT\n\n");
			goto rtuser_exit;
		}
		printf("\nATE_TXDONE_COUNT:: %08lx\n\n", (unsigned long)*wrq.u.data.pointer);

		//get SIGNAL_CONTEXT -----------------------------------------------------
		printf("\nrtuser::get SIGNAL_CONTEXT\n\n");
		{
			RT_SIGNAL_STRUC				*sp;

			memset(data, 0, sizeof(RT_SIGNAL_STRUC));
			strcpy(wrq.ifr_name, name);
			wrq.u.data.length = sizeof(RT_SIGNAL_STRUC);
			wrq.u.data.pointer = data;
			wrq.u.data.flags = RT_QUERY_SIGNAL_CONTEXT;
			ret = ioctl(socket_id, RT_PRIV_IOCTL, &wrq);
			if(ret != 0)
			{
				printf("\nrtuser::error::get SIGNAL_CONTEXT\n\n");
				goto rtuser_exit;
			}
			sp = (RT_SIGNAL_STRUC *)wrq.u.data.pointer;
			printf("\n===== SIGNAL_CONTEXT =====\n\n");
			printf("Sequence    = 0x%04x\n", sp->Sequence);
			printf("Mac.Addr    = %02x:%02x:%02x:%02x:%02x:%02x\n",
						sp->MacAddr[0], sp->MacAddr[1],
						sp->MacAddr[2], sp->MacAddr[3],
						sp->MacAddr[4], sp->MacAddr[5]);
			printf("CurrAP.Addr = %02x:%02x:%02x:%02x:%02x:%02x\n",
						sp->CurrAPAddr[0], sp->CurrAPAddr[1],
						sp->CurrAPAddr[2], sp->CurrAPAddr[3],
						sp->CurrAPAddr[4], sp->CurrAPAddr[5]);
			printf("Sig         = %d\n\n", sp->Sig);
		}

		//SSID, remove "set" string ----------------------------------------------
		memset(data, 0x00, 255);
		strcpy(data, "SSID=rtuser");
		strcpy(wrq.ifr_name, name);
		wrq.u.data.length = strlen(data)+1;
		wrq.u.data.pointer = data;
		wrq.u.data.flags = 0;
		ret = ioctl(socket_id, RTPRIV_IOCTL_SET, &wrq);
		if(ret != 0)
		{
			printf("\nrtuser::error::set SSID\n\n");
			goto rtuser_exit;
		}

rtuser_exit:
		if (socket_id >= 0)
			close(socket_id);

		if(ret)
			return ret;
		else
			return 0;
}
```

20 SingleSKU Example file (New feature for MT76XX)

20.1 2.4GHz example SingleSKU.dat

```
# Single SKU Max Power Table
# |CCK 1~11 | |   OFDM 6 ~ 54     | |          HT20 MCS 0 ~ 15               | |          HT40 MCS 0 ~ 15               |
ch1 23 23 23 23 21 21 21 21 21 21 20 20 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
ch2 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch3 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch4 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch5 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch6 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch7 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch8 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch9 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch10 23 23 23 23 22 22 22 22 22 22 20 20 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch11 23 23 23 23 19 19 19 19 19 19 19 19 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch12 23 23 23 23 19 19 19 19 19 19 19 19 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch13 23 23 23 23 19 19 19 19 19 19 19 19 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
ch14 23 23 23 23
```

Note: default SingleSKU profile path in driver is defined "/etc_ro/Wireless/RT2860AP/SingleSKU.dat"

For the detailed usage of SingleSKU in profile support, please refer to the MTK_SingleSKU_InProfile_User_manual.pdf and contact with MTK support windows.

20.2 5GHz example SingleSKU.dat

```
# Single SKU Max Power Table
#   OFDM 6 ~ 54          | |                    HT20 MCS 0 ~ 15                    | |               HT40 MCS 0 ~ 15                        | |            VHT80
MCS 0 ~ 9           |
ch36 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch38 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch40 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch42 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch44 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch46 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch48 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch52 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch54 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch56 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch58 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13
13 13 13 13 13 13
ch60 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch62 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch64 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch100 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
```

ch102 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch104 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 13 13 13 13
ch106 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13
13 13 13 13 13 13
ch108 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13
13 13 13 13 13 13
ch110 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13
13 13 13 13 13 13
ch112 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13
13 13 13 13 13 13
ch116 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch118 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch120 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch122 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch124 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch126 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch128 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch132 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 18 18 18 18
18 18 18 18 18 18
ch134 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 18 18 18 18
18 18 18 18 18 18
ch136 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 18 18 18 18
18 18 18 18 18 18
ch140 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch149 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch151 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch153 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch155 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch157 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch159 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch161 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch165 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch169 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
ch173 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18
End of Single SKU Table

Note: default SingleSKU profile path in driver is defined “/etc_ro/Wireless/RT2860AP/SingleSKU.dat”

For the detailed usage of SingleSKU in profile support, please refer to the MTK_SingleSKU_InProfile_User_manual.pdf and contact with MTK support windows.

21 How to Fix Data Rate

21.1 802.11n Data Rate Table

MCS index	Spatial streams	Modulation type	Coding rate	Data rate (Mbit/s)			
				20 MHz channel		40 MHz channel	
				800 ns GI	400 ns GI	800 ns GI	400 ns GI
0	1	BPSK	1/2	6.50	7.20	13.50	15.00
1	1	QPSK	1/2	13.00	14.40	27.00	30.00
2	1	QPSK	3/4	19.50	21.70	40.50	45.00
3	1	16-QAM	1/2	26.00	28.90	54.00	60.00
4	1	16-QAM	3/4	39.00	43.30	81.00	90.00
5	1	64-QAM	2/3	52.00	57.80	108.00	120.00
6	1	64-QAM	3/4	58.50	65.00	121.50	135.00
7	1	64-QAM	5/6	65.00	72.20	135.00	150.00
8	2	BPSK	1/2	13.00	14.40	27.00	30.00
9	2	QPSK	1/2	26.00	28.90	54.00	60.00
10	2	QPSK	3/4	39.00	43.30	81.00	90.00
11	2	16-QAM	1/2	52.00	57.80	108.00	120.00
12	2	16-QAM	3/4	78.00	86.70	162.00	180.00
13	2	64-QAM	2/3	104.00	115.60	216.00	240.00
14	2	64-QAM	3/4	117.00	130.00	243.00	270.00
15	2	64-QAM	5/6	130.00	144.40	270.00	300.00
16	3	BPSK	1/2	19.50	21.70	40.50	45.00
17	3	QPSK	1/2	39.00	43.30	81.00	90.00
18	3	QPSK	3/4	58.50	65.00	121.50	135.00
19	3	16-QAM	1/2	78.00	86.70	162.00	180.00
20	3	16-QAM	3/4	117.00	130.00	243.00	270.00
21	3	64-QAM	2/3	156.00	173.30	324.00	360.00
22	3	64-QAM	3/4	175.50	195.00	364.50	405.00
23	3	64-QAM	5/6	195.00	216.70	405.00	450.00
24	4	BPSK	1/2	26.00	28.80	54.00	60.00
25	4	QPSK	1/2	52.00	57.60	108.00	120.00
26	4	QPSK	3/4	78.00	86.80	162.00	180.00
27	4	16-QAM	1/2	104.00	115.60	216.00	240.00
28	4	16-QAM	3/4	156.00	173.20	324.00	360.00
29	4	64-QAM	2/3	208.00	231.20	432.00	480.00
30	4	64-QAM	3/4	234.00	260.00	486.00	540.00
31	4	64-QAM	5/6	260.00	288.80	540.00	600.00

21.2 2.4g

21.2.1 B only

```
iwpriv ra0 set FixedTxMode=CCK
iwpriv ra0 set WirelessMode=1          // 11b only
iwpriv ra0 set BasicRate=3             // 1, 2 Mbps
iwpriv ra0 set HtMcs=0                 // Please check Note-11b
iwpriv ra0 set SSID=11B_only_AP        // Restart AP
```

Note-11b:

HtMcs	**0**	**1**	**2**	**3**
Rate	1 Mbps	2 Mbps	5.5 Mbps	11 Mbps

21.2.2 G only

```
iwpriv ra0 set FixedTxMode=OFDM
iwpriv ra0 set WirelessMode=4          // 11g only
iwpriv ra0 set BasicRate=351           // 1, 2, 5.5, 11, 6, 12, 24 Mbps
iwpriv ra0 set HtMcs=0                 // Please check Note-11g
iwpriv ra0 set SSID=11G_only_AP        // Restart AP
```

Note-11g:

HtMcs	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Rate	6 Mbps	9 Mbps	12 Mbps	18 Mbps	24 Mbps	36 Mbps	48 Mbps	54 Mbps

21.2.3 N only

```
iwpriv ra0 set FixedTxMode=HT
iwpriv ra0 set WirelessMode=6          // 2.4g 11n only
iwpriv ra0 set BasicRate=15            // 1, 2, 5.5, 11 Mbps
iwpriv ra0 set HtMcs=0                 // Please check Note-11n
iwpriv ra0 set HtGi=0
iwpriv ra0 set HtBw=0
iwpriv ra0 set SSID=11GN_only_AP       // Restart AP
```

Note-11n:

HtMcs=<0-15> + HtGi=<0-1> + HtBw=<0-1>

Please check all possible combination of above set in Section 21.1.

21.2.4 B/G/N mixed

```
iwpriv ra0 set FixedTxMode=HT
iwpriv ra0 set WirelessMode=9          // 11bgn mixed
iwpriv ra0 set BasicRate=15            // 1, 2, 5.5, 11 Mbps
iwpriv ra0 set HtMcs=0                 // Please check Note-11n
iwpriv ra0 set HtGi=0
iwpriv ra0 set HtBw=0
iwpriv ra0 set SSID=11BGN_mixed_AP     // Restart AP
```

Note-11n:

HtMcs=<0-15> + HtGi=<0-1> + HtBw=<0-1>

Please check all possible combination of above set in Section 21.1.

21.3 5g

21.3.1 A only

```
iwpriv ra0 set FixedTxMode=OFDM
iwpriv ra0 set WirelessMode=2          // 11a only
iwpriv ra0 set BasicRate=336           // 6, 12, 24 Mbps
```

iwpriv ra0 set HtMcs=0 // Please check Note-11a
iwpriv ra0 set SSID=11A_only_AP // Restart AP

Note-11a:

HtMcs	0	1	2	3	4	5	6	7
Rate	6 Mbps	9 Mbps	12 Mbps	18 Mbps	24 Mbps	36 Mbps	48 Mbps	54 Mbps

21.3.2 N only

iwpriv ra0 set FixedTxMode=HT
iwpriv ra0 set WirelessMode=11 // 5g 11n only
iwpriv ra0 set BasicRate=336 // 6, 12, 24 Mbps
iwpriv ra0 set HtMcs=0
iwpriv ra0 set HtGi=0
iwpriv ra0 set HtBw=0
iwpriv ra0 set SSID=11AN_only_AP // Restart AP

Note-11n:
HtMcs=<0-15> + HtGi=<0-1> + HtBw=<0-1>
Please check all possible combination of above set in Section 21.1.

21.4 11ac

21.4.1 VHT Fixed Rate iwpriv command

21.4.1.1 fpga_on

Description: Turn on or off VHT fixed rate
Value:

iwpriv rai0 set fpga_on=6

0: Disable
6: Enable

21.4.1.2 dataphy

Description: PHY mode configuration
Value:

iwpriv rai0 set dataphy=4

0 = CCK
1 = OFDM
2 = HT-MM
3 = HT-GF
4 = VHT

21.4.1.3 databw

Description: Bandwidth configuration
Value:

iwpriv rai0 set databw=2

0 = 20M
1 = 40M
2 = 80M

21.4.1.4 datamcs

Description: MCS configuration
Value:

iwpriv rai0 set datamcs=24

Note
bit[3:0] stands for Modulation Coding Scheme (MCS)

Range: 0 - 9

bit[6:4] stands for Number of Spatial Stream (NSS)

0: 1SS
1: 2SS

Example:
datamcs=24 → 2SS MCS8
24 (dec) = 0x18 = b'0001,1000
bit[6:4] = b'001 = 1 (dec) → 2SS
bit[3:0] = b'1000 = 8 (dec) → MCS8

1SS & 2SS MCS Rate mapping table:

1SS

MCS Index	Modulation	Value (Dec)
0	BPSK	0
1	QPSK	1
2	QPSK	2
3	16-QAM	3
4	16-QAM	4
5	64-QAM	5
6	64-QAM	6
7	64-QAM	7
8	256-QAM	8
9	256-QAM	9

2SS

MCS Index	Modulation	Value (Dec)
0	BPSK	16
1	QPSK	17
2	QPSK	18
3	16-QAM	19
4	16-QAM	20
5	64-QAM	21
6	64-QAM	22
7	64-QAM	23
8	256-QAM	24
9	256-QAM	25

21.4.1.5 datagi

Description: Guard Interval configuration

Value:

iwpriv rai0 set datagi=0

0 = Short GI,
1 = Long GI

21.4.2 VHT Fixed Rate example

iwpriv rai0 set WirelessMode=14
iwpriv rai0 set fpga_on=6 // Enable VHT fixed rate
iwpriv rai0 set dataphy=4 // VHT
iwpriv rai0 set databw=2 // 80MHz
iwpriv rai0 set datagi=0 // SGI
iwpriv rai0 set datamcs=25 // 2SS MCS9

The following 802.11ac rate table is from http://www.revolutionwifi.net/.

802.11ac OFDM Data Rates

MCS	Modulation	Bits per Symbol	Coding Ratio	20-MHz 800ns	20-MHz 400ns	40-MHz 800ns	40-MHz 400ns	80-MHz 800ns	80-MHz 400ns
1 Spatial Stream				**Data Rate (Mbps)**					
MCS 0	BPSK	1	1/2	6.5	7.2	13.5	15.0	29.3	32.5
MCS 1	QPSK	2	1/2	13.0	14.4	27.0	30.0	58.5	65.0
MCS 2	QPSK	2	3/4	19.5	21.7	40.5	45.0	87.8	97.5
MCS 3	16-QAM	4	1/2	26.0	28.9	54.0	60.0	117.0	130.0
MCS 4	16-QAM	4	3/4	39.0	43.3	81.0	90.0	175.5	195.0
MCS 5	64-QAM	6	2/3	52.0	57.8	108.0	120.0	234.0	260.0
MCS 6	64-QAM	6	3/4	58.5	65.0	121.5	135.0	263.3	292.5
MCS 7	64-QAM	6	5/6	65.0	72.2	135.0	150.0	292.5	325.0
MCS 8	256-QAM	8	3/4	78.0	86.7	162.0	180.0	351.0	390.0
MCS 9	256-QAM	8	5/6	N/A	N/A	180.0	200.0	390.0	433.3
2 Spatial Streams				**Data Rate (Mbps)**					
MCS 0	BPSK	1	1/2	13.0	14.4	27.0	30.0	58.5	65.0
MCS 1	QPSK	2	1/2	26.0	28.9	54.0	60.0	117.0	130.0
MCS 2	QPSK	2	3/4	39.0	43.3	81.0	90.0	175.5	195.0
MCS 3	16-QAM	4	1/2	52.0	57.8	108.0	120.0	234.0	260.0
MCS 4	16-QAM	4	3/4	78.0	86.7	162.0	180.0	351.0	390.0
MCS 5	64-QAM	6	2/3	104.0	115.6	216.0	240.0	468.0	520.0
MCS 6	64-QAM	6	3/4	117.0	130.0	243.0	270.0	526.5	585.0
MCS 7	64-QAM	6	5/6	130.0	144.4	270.0	300.0	585.0	650.0
MCS 8	256-QAM	8	3/4	156.0	173.3	324.0	360.0	702.0	780.0
MCS 9	256-QAM	8	5/6	N/A	N/A	360.0	400.0	780.0	866.7

22 Q&A

22.1 Why does WPAPSK not work?

Please make sure the parameter **"DefaultKeyID" is set to 2** in the configuration file.

22.2 How to switch driver to operate in 5G band?

Please make sure the IC supports 5G band.
Also, please configure the WirelessMode and Channel correctly.

22.3 How do I check my channel list?

Please check CountryRegion or CountryRegionABand.

22.4 How can I know the version of current WLAN Driver?

Please use the following command.
iwpriv ra0 show driverinfo

22.5 Can SoftAP support Antenna diversity?

No, SoftAP do not support antenna diversity even EEPROM has set antenna enabled.

22.6 DFS Test example

Case 1: Band 2 & 3 select one channel for test
Test Condition:
Run 30% throughput between STA and AP.

DFS Debug command:
iwpriv ra0 set RadarDebug=0x10

DFS CE certification setting in the profile:
IEEE80211H=1
DfsOutdoor=0
RDRegion=CE
CountryCode=GB

Result:
All major test items are all passed.

Case 2: Band 2 & 3 select one channel for test.

Test condition:
Run video stream throughput between STA and AP. (Set AP Fix Tx Rate to MCS0)
Bandwidth setting 20MHz and 20/40MHz Auto.

DFS Debug command:
iwpriv ra0 set RadarDebug=0x10

DFS FCC certification setting in the profile:
IEEE80211H=1
DfsOutdoor=0
RDRegion=FCC
CountryCode=US

Result:
When Radar signal run in 5498~5502MHz, Radar type 3 & 4 fail in BW 40MHz test.
Radar type 1 fail in BW 20MHz test, Recommend to make the Radar signal run in 5495~5525MHz with BW 40MHz test. In 5494~5506MHz in BW 20MHz test. All major test items are all passed.

Case 3: Detect DFS signal without move channel. (For Lab testing)
Command Example:

iwpriv ra0 set Debug=3
iwpriv ra0 set Channel=100
iwpriv ra0 set RadarDebug=0x10
iwpriv ra0 set ChMovTime=2
iwpriv ra0 set DfsSwDisable=0

Result:
When Radar signals run in channl 100, the AP will display DFS detected information on the console.

DFS deteched consloe log may look like below:

> **DFS HW check channel = 0x4**
> **T= XXXXX W= XXX detected by ch 2**

22.7 TX & RX performance is always unbalance

When encounter TX & RX performance unbalance issue during Wi-Fi performance test, please check the TxBurst option is off or on. When TxBurst is on, the TX packets will have higher priority than RX packets. In the result, the WLAN TX performance will be higher than RX. This problem usual appears in Fast Ethernet + WLAN solution. GiGaBit Ethernet + WLAN solution doesn't have such problem.

How to turn off TxBurst?
By profile:
TxBurst=0
By iwpriv command:
iwpriv ra0 set TxBurst=0

22.8 Why can’t I configure a SSID containing comma “,”?

Please modify your code as follows.

```
==========
INT RTMPAPPrivIoctlSet(
        IN RTMP_ADAPTER *pAd,
        IN RTMP_IOCTL_INPUT_STRUCT *pIoctlCmdStr)
{
        PSTRING this_char;
        PSTRING value;
        INT Status = NDIS_STATUS_SUCCESS;

        while ((this_char = strsep((char **)&pIoctlCmdStr->u.data.pointer, "\0")) != NULL)
        {
                if (!*this_char)
                        continue;

                if ((value = strchr(this_char, '=')) != NULL)
                        *value++ = 0;
```

22.9 Why throughput is low when using 1SS to send traffic with legacy rate or MCS0-7?

Using 2SS to send traffic with legacy rate and MCS0-7 is our design by default. If you intend to change from 2SS to 1SS, please use TC instead of TSSI.

22.10 TGn 4.2.10 failed. Why does DUT not send MC traffic?

4.2.10 Group traffic with WPA2-PSK Only Mode and WPA/WPA2-PSK Mixed Mode
If this item fails, please turn off IGMP Snooping first.

22.11 TGn 4.2.29 failed. Why the performance cannot reach the criteria?

Please make sure that the following items are correctly configured.
<Profile>
TxPreamble=1
PktAggregate=0

<Driver Config>
-CONFIG_RA_NETWORK_WORKQUEUE_BH=y
+CONFIG_RA_NETWORK_TASKLET_BH=y

<Kernel Config>
Please check items in Networking Option & Core Netfilter in your kernel config. Remove those you do not use or know.